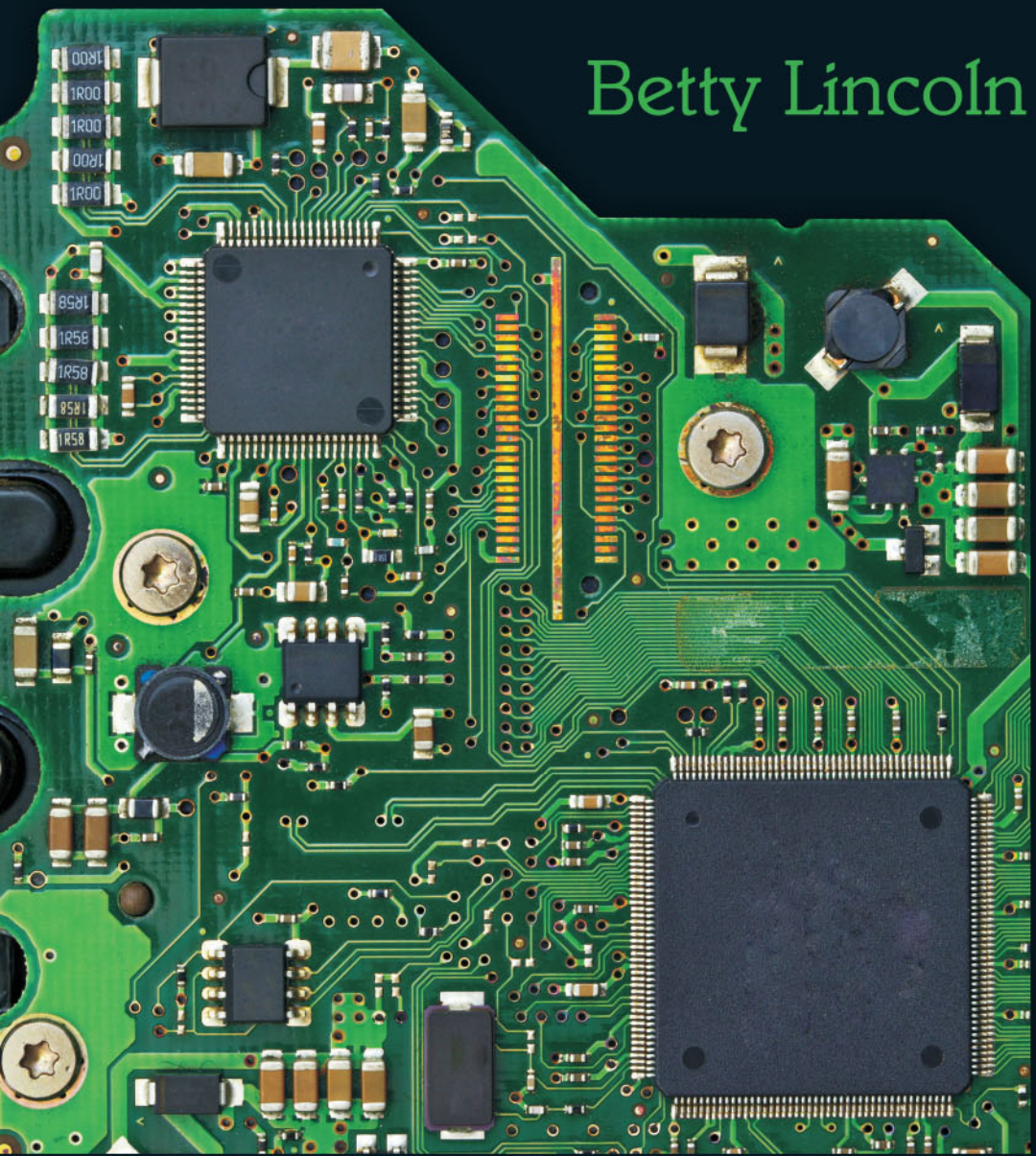
Digital Electronics

Betty Lincoln



ALWAYS LEARNING

PEARSON

SHOUT
SHARE
IT

Copyright © 2014 Dorling Kindersley (India) Pvt. Ltd.
Licensees of Pearson Education in South Asia

No part of this eBook may be used or reproduced in any manner whatsoever without the publisher's prior written consent

This eBook may or may not include all assets that were part of the print version. The publisher reserves the right to remove any material in this eBook at any time.

ISBN 978-93-325-2229-9
eISBN 978-93-325-4072-9

Head Office: A-8(A), Sector 62, Knowledge Boulevard, 7th Floor, NOIDA 201 309, India
Registered Office:11 Local Shopping Centre, Panchsheel Park, New Delhi 110 017, India

Dedicated to the Glory of God

The fear of the Lord is the beginning of wisdom

Proverbs 9: 10 (The Holy Bible)

CONTENTS

Preface xv
Acknowledgement xvii

CHAPTER 1 **Digital Systems** 1
Digital transmission—3
Multiple Choice Questions 6
Descriptive Questions 7

CHAPTER 2 **Numbering Systems** 9
Introduction—9
Decimal number system—10
Binary numbering system—10
Octal numbering system—10
Hexadecimal numbering system—11
Conversion of numbers from one base into another—12
Arithmetic in number system—15
Binary subtraction—17
Negative binary numbers—18
Binary multiplication—21
Binary division—22
Number codes—23
Gray code—27
Alphanumeric codes—30
Parity method for error detection—33
Multiple Choice Questions 40
Descriptive Questions 42

CHAPTER 3 **Logic Gates** 45
Introduction—45
Positive and negative logic—45
AND gate—46

OR gate—48
NOT gate—49
Universal gates—50
NAND as a universal gate—51
Operation of a NAND gate—51
NOR gate—53
Operation of NOR gate—54
The EXCLUSIVE-OR and EXCLUSIVE-NOR gates—55
Operation of XOR gate—56
EX-NOR gate (EXNOR/XNOR)—56
IEEE symbols—58
Additional examples—59
Multiple Choice Questions 70
Descriptive Questions 72

CHAPTER 4 **Boolean Algebra** **73**
Fundamental symbols—Boolean algebra—73
Boolean algebraic properties—73
De Morgan's theorem—75
Logic axioms—76
Minterms and maxterms—79
Additional examples—81
Multiple Choice Questions 88
Descriptive Questions 89

CHAPTER 5 **Karnaugh Mapping** **91**
Karnaugh map for two variables—91
Karnaugh map for three variables—93
Karnaugh map for four variables—93
Karnaugh map for five variables—94
Rules in Karnaugh mapping—101
Additional examples—107
Summary 117
Multiple Choice Questions 117
Descriptive Questions 118

CHAPTER 6 **Implementation of Universal Gates** **121**
NAND implementation—121

NOR as a universal gate—122
Equivalent gates—123
NAND and NOR implementation to Boolean functions—124
NAND-AND and AND-NOR implementation—127
NOR-OR implementation and OR-NAND implementation—127
Multiple Choice Questions 128
Descriptive Questions 130

CHAPTER 7 **Integrated Circuits and Logic Families** **131**
Introduction—131
Classification of ICs—132
Classification based on size—132
Common digital IC terminology—132
Characteristics of logic families—133
Semiconductor diodes and transistors—135
Semiconductor diodes—137
Semiconductor transistors—139
Logic families—144
IC packaging—157
Troubleshooting in digital devices—157
Summary 159
Multiple Choice Questions 159
Descriptive Questions 161

CHAPTER 8 **Combinational Logic Circuits** **163**
Combinational circuits—163
Binary adders—163
Half adder—164
Full adder—165
Addition of more than 2 bits by full adder—167
Parallel adder–subractor—171
Multiplexers—172
2 × 1 Multiplexer—173
4 × 1 Multiplexer—173
Quad 2 × 1 multiplexer—174
Implementation of Boolean functions using multiplexers—175
Demultiplexer—177
Code convertors—180

Decoders—181
Encoders—184
Multiple Choice Questions 186
Descriptive Questions 188

CHAPTER 9 **Flip-Flops** **189**
Latch—192
Clocked SR latch—195
Clock pulses—197
D latch—200
Master–slave D flip-flop—202
Edge-triggered D flip-flop—202
JK flip-flop—203
Master–slave JK flip-flop—208
T flip-flop—209
Conversions of flip-flops—210
Applications of flip-flops—216
Multivibrators—217
Bistable multivibrator—217
Monostable multivibrator—219
Astable multivibrator—221
Schmitt trigger—223
IC 555 timer—224
Multiple Choice Questions 227
Descriptive Questions 229

CHAPTER 10 **Registers** **231**
Shift registers—231
Bidirectional shift registers—242
Universal shift register—244
Shift registers applications—244
Multiple Choice Questions 252
Descriptive Questions 254

CHAPTER 11 **Counters** **257**
MOD-3 counter—258
MOD-16 or MOD-4 bit counter—259
Down counters—261

Designing of ripple counter divided by *N*—261
Decade counter or MOD-10 ripple counter—262
BCD counter—262
Decoding a counter—264
UP/DOWN counter—264
Parallel counter (or) synchronous counter—265
Decade synchronous counter—267
Synchronous UP/DOWN counter—268
IC counters—269
Applications of counters—270
Additional examples—270
Multiple Choice Questions 277
Descriptive Questions 279

CHAPTER 12 **Memory** **281**
Dynamic random access memory (DRAM)—287
General DRAM modules—289
DRAM memory modules—290
Read only memory—291
Programmable read only memory—292
EPROM and EEPROM—293
Flash memory—293
Programmable array logic—294
Programmable logic array—296
Summary 297
Multiple Choice Questions 298
Descriptive Questions 299

CHAPTER 13 **Display Devices** **301**
Cathode ray tubes (CRTs)—301
Liquid crystal displays (LCD)—304
Liquid crystals on silicon (LCOS)—310
Light emitting diodes (LED)—312
7-Segment display—314
Introduction to organic light emitting diodes (OLED)—317
Plasma displays—321
Digital light processing—323
E-books/e-ink—325

Multiple Choice Questions 326
Descriptive Questions 329

CHAPTER 14 **Convertors** **331**
Analog-to-digital conversion—331
ADC process—334
Accuracy of A/D conversion—335
Types of A/D converters—336
Digital ramp ADC—336
Successive approximation ADC—338
Flash A/D converter—340
Sigma-delta A/D converter—342
Dual slope convertor—343
Digital-to-analog convertors—344
Applications of DAC—351
Multiple Choice Questions 351
Descriptive Questions 353

CHAPTER 15 **Computer Fundamentals** **355**
Introduction—355
Computer characteristics—355
Evolution of computers—356
Generations of computers—357
Classification of computers—360
Components of digital computers—364
Input–output devices—371
Output devices—374
Computer software—377
Computer languages—378
Multiple Choice Questions 383
Descriptive Questions 384

CHAPTER 16 **Electronic Exercises** **385**
CMOS gate – CD 4001 NOR—385
Modulus counter – JK flip-flops—388
Modulus counter – D flip-flops 7474—391
Modulus counters – JK flip-flops – synchronous—393
Modulus counter – decade counter (7490)—395

Study of 8-bit shift register (7495)—398
Study of up/down counter—400
Design of counters—402
Study of binary adder and subtractor circuits—403
Study of ALU and BCD adder circuits—405
Study of DEMUX circuits—407
Study of digital encoder/decoder circuits—410
Study of RAM—413
A/D converter—415
D/A converter—417
Study of MUX—419

Index 423

PREFACE

Digital electronics deals with the designing of digital circuits. Digital circuits form the backbone of communication systems, recording devices and many more. Digital electronics is a fast expanding field with scope for significant development. This book discusses the basics of designing circuits and the functioning of intricate components of a digital circuit. It presents the fundamental concepts and serves as an excellent guide for an introductory course on digital electronics. The simple yet effective treatment of theory is reinforced by a practical approach that caters to the need of students of computer science, electronics and communication engineering, and computer engineering specializations.

Chapter 1 introduces the student to digital and analog signals, and spells out the advantages of digital signals over analog signals and digital signal transmission. It deals at length with error occurrence and error detection during digital transmission.

Chapter 2 discusses the hexadecimal, decimal, octal and binary numbering systems. It illustrates the conversion of one number system to the other, the arithmetic in number systems, and negative binary numbers with ample worked out problems. The chapter gives an insight into alphanumeric codes such as the American standard code for information interchange (ASCII), the extended binary coded decimal interchange code (EBCDIC), the Gray code, the Hollerith code, and the Excess–3 codes. It also expounds on the methods of error detection during transmission using the Hamming code, the cyclic redundancy check code and the repetition code.

Chapter 3 brings out the concept of logic gates. It explains the various types of logic gates, lists their IEEE symbols and proffers a sufficient number of worked-out problems that depict the circuitry of each gate. It also elaborates on the use of NAND and NOR as universal gates.

Chapter 4 explores the fundamentals of Boolean algebra, its properties and axioms. It provides several examples that establish a correlation between the Boolean expression and its logic circuit. The chapter delves into the procedure for reduction of logic expressions using algebraic axioms.

Chapter 5 covers the Karnaugh Mapping tool used to reduce Boolean expressions. It explains the sum of the product form (SOP) and product of the sum form (POS) with examples and reveals the steps to be followed during the transition of the forms.

Chapter 6 outlines the implementation of the universal gates NAND and NOR for representing the basic logic gates, AND, OR and NOT.

Chapter 7 deals with the common integrated circuits and the digital logic families. Since a sound knowledge of circuit functioning is essential to comprehend this chapter, it begins with an introduction to semiconductor diodes and transistors and proceeds to unravel the working of logic families. The chapter reviews classification of the logic families, the common digital IC terminology and their characteristics while arguing about its

merits and demerits. The chapter also exposes students to the techniques of trouble shooting in digital devices and circuits.

Chapter 8 probes the design and analysis of combinational circuits. It examines the basic components of combinational circuits such as half adder, full adder, parallel adder and subtractor using sound examples. It acquaints students with multiplexers, demultiplexers, code convertors, decoders and encoders, which are frequently used for designing digital circuits.

Chapter 9 elucidates sequential circuits, going into the details of the various types of latches and flip-flops. It delineates the types of clock pulses, the arrangement of flip-flops using gates and the effect of edge triggering on the flip-flops. The chapter also investigates the conversions of flip-flops, their applications, the multivibrator and its types, and the IC555 timer.

Chapter 10 deals with components such as registers that form the basic building block of digital circuits. It discusses the types of shift register and their applications with examples

Chapter 11 familiarizes students with counters, which form an important component of digital systems. The chapter takes a close look at the types of counters and their functions and provides an overview of their various applications.

Chapter 12 focuses on memory devices such as random access memory (RAM) and its types, read only memory (ROM), programmable read only memory (PROM) and its variations, programmable array logic (PAL) and programmable logic array (PLA).

Chapter 13 gives a vivid explanation of display devices, which are indispensable components of digital circuits. It sheds light on the various forms of display devices and the developments made in these devices, with emphasis on the construction, working principle, merits and demerits of the CRT, LCD, LCOS, LED, the seven-segment display, the OLED, plasma displays, digital light processing, e-books and e-ink.

Chapter 14 analyzes the important methods of converting signals from analog to digital and vice versa and highlights them with ample worked-out examples. The chapter also spells out the characteristics and applications of ADC and DAC.

Chapter 15 presents the fundamentals of computers in a nutshell, beginning with a brief note on the evolution of computers. The chapter outlines the classification of computers and the components of digital computers. It also includes an introduction to commonly used computer software and computer languages.

Chapter 16 compiles a series of experiments that can be carried out to enhance one's knowledge on digital electronic circuits, applying them to practical scenarios.

The concepts laid down in each chapter are augmented by numerous worked-out examples that showcase real-life applications to enable students clearly understand the functioning of digital systems. In addition, chapter-end multiple choice questions and descriptive questions serve as an effective self-evaluation tool for the student.

Betty Lincoln

ACKNOWLEDGEMENT

At the outset, I thank God Almighty for His presence and grace that sustained me and helped me to accomplish the onerous task of writing this book. I would like to mention and thank a few people who were instrumental in helping me to complete this book project.

I am grateful to the Pearson publishing team for their efforts in bringing this project to fruition. I specially thank Ms R. P. Mathivathani, whose initial suggestions that helped me shape this book and Ms Vijayapritha, who contributed immensely in the later stages. I thank Mr. Ramesh and his production team for their meticulous proof reading for setting the book's contents in an excellent layout.

I am indebted to my mom Mrs Geetha Lincoln and dad Mr V. A. Lincoln for their constant support and prayers and my sister Lorraine for her unconditional love and advice at times when I was feeling low.

I thank my friends, Sumitha for her care and faith in me and Deepa, who has always been a driving force at every step. I am obliged to Mrs Meryn Selvanayagam who has been a pillar of strength and motivation. I thank her for correcting the grammatical errors. I appreciate the valuable inputs provided by Ms Divya, which enhanced the overall value of this book.

I am obliged to my teachers who were instrumental in helping me develop an interest towards electronics. I thank all my beloved students for their help and love.

I express my gratitude to Shri V. R. Venkataachalam, our beloved chancellor, and the management of the Sri Ramachandra University for giving me an opportunity to serve this esteemed organization. I am honoured by their support and encouragement.

Last but not least, I thank T. S. Lokeswari, Head, Department of Biomedical Sciences, and Dr Solomon Paul, Head, Department of Human Genetics, for building my confidence and for their steady guidance in shaping my professional career.

Betty Lincoln

DIGITAL SYSTEMS

Digitalization has proved to be indispensible to the world. Digital camera, digital discs, digital computers—everything around the world has become digital. It is impossible to comprehend a world without digital devices. It is common to see teenagers and young children with hi-tech gadgets at their disposal.

Digital systems basically work by means of signals. Signals are of two types—analog and digital. Analog signals are those which can take a range of voltages. Values of analog signals are found to change with time. Physical quantities such as pressure, velocity, sound, current flow are examples of analog signals that are converted to digital signals by means of devices called transducers. Transducers are devices that convert one form of energy to another form.

On the other hand, digital signals do not vary with time but take a definite value at any given instant of time. The role of digital circuits in both personal and professional arenas makes it imperative that every individual acquires a good knowledge of the basic concepts and gets acquainted with digital circuits.

The introduction of semiconductor technology has given an unprecedented boost to digital circuits. The complexity of integrated circuits is increased by increasing the number of transistors (as given by Moore's law).

In general, a digital system can have n defined states. For a binary system, $n = 2$. It can permit only two states—HIGH or LOW. A value of 1 represents "HIGH" state while "LOW" state is represented by a value of "0". The devices that use these two states are referred to as two state devices. These devices can be operated only in the ON or OFF states and not in any intermediate state. A semiconductor diode is an example of a two state device. When the diode is forward biased, it conducts and is hence in the "ON" state, when it is reverse biased, it does not conduct and hence is in the "OFF" state. Thus, digital circuits operated by diodes and transistors can switch between ON and OFF state swiftly. This and the fact that they are cheap for mass production make them strong candidates for the construction of digital circuits. While digital signals can be transmitted over long distances effectively, they do have certain disadvantages. A combination of 0's and 1's make a digital signal complicated by achieving several different combinations. However, it would be very difficult to interpret these signals. In addition, digital systems can be manipulated to provide a wide range of options.

Digital systems are more prone to quality loss. It can be corrected by using error detection techniques. These techniques help in removing disturbances in the digital

signal, thereby improving its quality. Digital signals can be classified into unipolar and bipolar. A unipolar digital signal is one that has only single polarity. It can have either positive or negative voltage that switches from zero to its corresponding voltage. They are represented in Figure 1.1 (a) and (b).

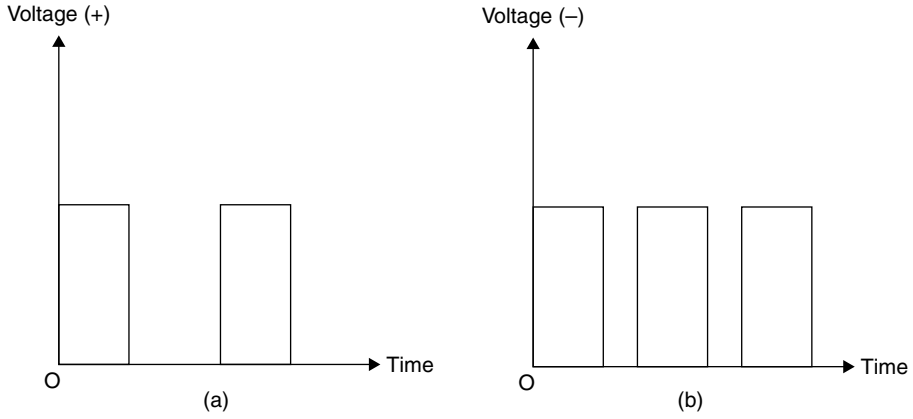


Fig.1.1 **(a)** Positive unipolar digital signal and **(b)** Negative unipolar digital signal

The bipolar digital signal on the other hand is one that fluctuates between a positive and negative voltage with time and is represented as in Figure 1.2.

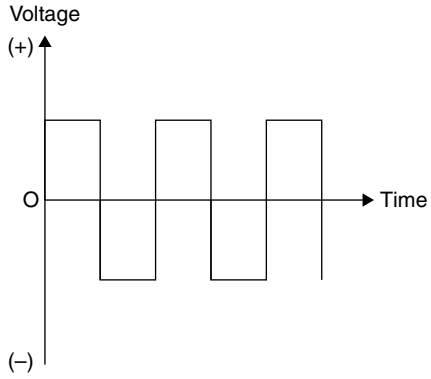


Fig.1.2 Representation of bipolar signals

The timing of the digital signals are highly essential for the precise functioning of digital circuits. The waveform repeats at regular time interval and hence are referred as periodic clock pulses. The time period T of a given clock pulse is the length of time from the leading edge of one pulse to leading edge of the next pulse. It is also described as the distance between the trailing edge of one pulse to the trailing edge of the next pulse. The clock frequency is given as the reciprocal of time period and is represented as

$$f = \frac{1}{T}$$

If the waveforms are rectangular, then the ratio is $\frac{t_1}{T}$. The mark-space ratio of a square wave is $\frac{t_1}{t_2}$. The value of the mark-space ratio is 0.5. The waveforms represented in the Figure 1.2 does not rise and fall instantly. It takes time to rise to the desired level and to fall back to zero. The time taken by the waveform to reach to its peak value is the rise time and the time taken by the waveform to fall back from its peak value is fall time. A typical digital signal may have the rise and fall time in the range of few nano-seconds. The mark-space ratio, rise and fall time are represented in Figures 1.3 and 1.4 where t_r = rise time and t_f = fall time.

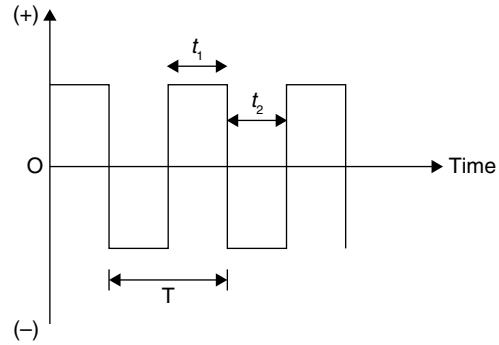


Fig.1.3 Mark-space ratio

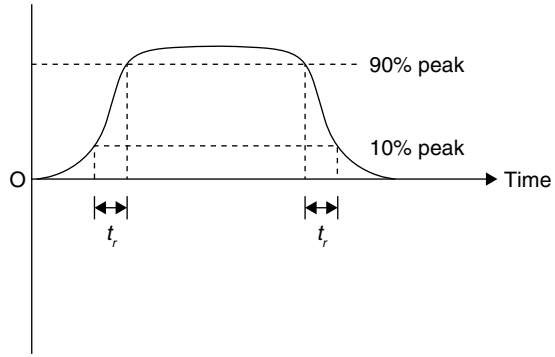


Fig.1.4 Rise time and fall time

DIGITAL TRANSMISSION

The service of providing long distance transmission of electrical signals is known as tele-communication. The input data which is in the analog form which is a characteristic of frequency of voltage, amplitude, frequency, phase etc is converted into digital signal that

are discrete in nature. The conversion and transmission of the input information requires the following components. Digital transmission requires a transmitter, to transmit the input information and a receiver to receive the desired output. A transmitter converts the input data into signal form suitable for transmission. It feeds energy into communication channels that are later converted. For example, a telephone receiver converts voice into electric current. A digital transmission process is represented in Figure 1.5.

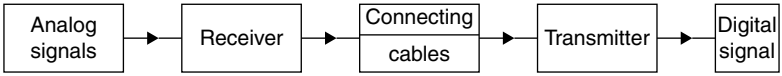


Fig.1.5 Digital Transmission process

Receiver at the other end, receives energy from medium. It converts received signal into the desired form that the user requires. For example, telephone also converts current into voice.

The transmitter and receiver are connected through a communication channel. A communication cable could be a pair of copper wires, coaxial cable or light in case of optical fiber. The digital signals are preferred over analog signal for transmission since the former is free from distortion. Long distance analog communications require repeaters between the source and destination that are capable by restoring analog signals to its original form. The restored signals are not completely eliminated of distortion. This is because the noise and interference caused by external sources are not completely eliminated. The number of repeaters depends on the signal data and this deteriorates the quality of analog signals. For example, analog communication is a song being taped in a cassette recorder (Figure 1.5).

Digital long distance communications receives the original data at the next segment of transmission. These have the repeaters being replaced by regenerators that recover original data sequence and again retransfers to the next immediate level. Owing to the factor that the long distance digital communications consume less power, longer distances and lower system cost, they are controlled and are preferred.

The main fundamental issue with digital transmission is the fact that the delay in transmission. This time delay varies with the material of the communication channel. The propagation speed of signal is given as

$$V = \frac{C}{V_\varepsilon} \quad \text{where} \quad C = 3 \times 10^8 \text{ m/s}$$

and V_ε = speed of light in the medium provided with dielectric constant of medium ε is greater than 1. For copper wire the propagation speed is 2.3×10^8 m/s and in an optical fiber, the propagation speed is 2×10^8 m/s.

Error Detection and Error Correction in Digital Transmission

The transmission of digital signals introduces errors. Since most applications require reliability in the transmission process, the transmitted data must be free of errors. But there are few applications that posses tolerable level of errors, for example, video and

audio transmission. Therefore error control is an essential step in the process of data transmission. The error control ensures data transmitted to a certain level of accuracy with tolerable levels of errors.

Error control can be carried out in two steps—error detection and error correction. There are many techniques that are used for error detection and correction namely Hamming code, parity check, redundancy check etc. The information of the user is sent to the encoder. The encoder sends its output to the channel. All inputs to the channel follow a specific pattern or condition. Any change from this pattern is considered as error. The channel output is given to the pattern check block that checks for parity or redundancy check and delivers the user information. If an error has occurred, then it sets for an alarm. The process is represented in Figure 1.6.

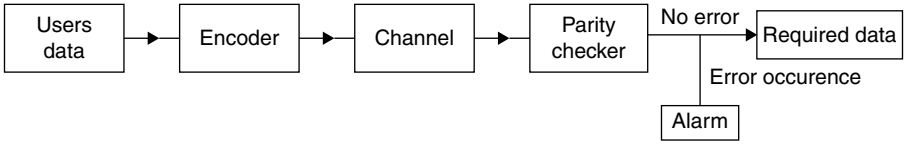


Fig.1.6 Error detection and correction

Some of the error detection techniques are explained elaborately in Chapter 2 of this book.

Advantages and Disadvantages of Digital Signal Transmission

The digital transmission has few advantages that are listed below.

- The system noise is controlled to a great extent by designing of the terminal. The noise is independent of the length of the system.
- The usage of transmission channels can be lowered by increasing the speed of the digital device. This can be achieved by constructing circuits using integrated chips, VLSI etc. This adaptation technique cuts down the cost of each channel and thereby the cost of the complete system.
- Digital transmission is flexible to the user in terms of multiplexing digital data, video (or) voice data transmission but this is not achievable in analog transmission.
- The digital transmission systems requires lesser number of channels to obtain a given digital data and is more efficient than analog systems.
- Digital systems are insensitive to overcrowding of data during transmission process and can hold data upto their full capacity.
- The digital switches have inbuilt input and output terminals that perform at the speed of multiplexers. Hence a digital transmission system provides efficient interfacing between switching systems whereas the analog system require demultiplexers and multiplexers at the rate of the switching terminals.
- The alteration of a digital system is independent of frequency specifically with the usage of fiber optic transmission.

- Sometimes the accumulation of bits across the digital system may result in errors and data cannot be recovered till the error is detected and correction is performed.
- Digital systems are cost effective compared to analog transmission system but the metallic media used in transmission lines are expensive and may increase the per channel cost. The cost can be cut down by replacing the connecting cables with optic fibres.

MULTIPLE CHOICE QUESTIONS

1. Telecommunication refers to the transmission of
   (a) Sound waves as voice (b) Digital waves
   (c) Electrical signals (d) None of the above
2. The Input data to be transmitted through telecommunication cannot be in the form of
   (a) Binary bits (b) Frames
   (c) Sound signals (d) Blocks
3. Which of the following medium is not as a channel for digital transmission?
   (a) Coaxial cable (b) Optical fiber
   (c) Radio frequency (c) Air
4. An analog signal or waveform is a continuously varying signal characterized by
   (a) Amplitude only (b) Frequency only
   (c) Both amplitude and frequency (d) None of the above
5. Analog signals have _____ values.
   (a) 10 (b) Infinite
   (c) Finite (d) None of the above
6. Which of the following are used for conversion of signals?
   (a) Modems (b) Multiplexers
   (c) Transformers (d) Transducers
7. Computer is a source for
   (a) Analog signals (b) Sinusoidal waves
   (c) Digital signals (d) Triangular waves
8. The voltage in digital signaling is always either positive or negative similar to dc current.
   (a) Unipolar (b) Bipolar
   (c) Parallel (d) Serial
9. Which of the following is an example for digital signals?
   (a) Voice (b) Temperature
   (c) Binary bits (d) Pressure

Answers

1. (c) 2. (c) 3. (d) 4. (c) 5. (b) 6. (a) 7. (c) 8. (a) 9. (c)

DESCRIPTIVE QUESTIONS

1. Differentiate Analog signals and Digital signals.
2. Mention the disadvantages of Analog Transmission.
3. Explain the types by which data can be transmitted in digital devices.
4. What are unipolar and bipolar digital signals?
5. Mention the advantages and disadvantages of digital transmission .
6. Discuss the various steps involved in digital transmission with a neat diagram.
7. Mention the applications of digital transmission process.

NUMBERING SYSTEMS

INTRODUCTION

Mathematical computations have been performed using decimal numbers 0–9 for ages. All the data from the commercial bills to the very countdown timing for the launch of a satellite is done using decimal digits. Owing to the extensive usage of decimal number system, the other number systems are overshadowed. However, these decimal representations are restricted to manual counting, while computerized counting requires an alternative method—binary system. Computers cannot operate 10 digits as its processor's circuitry has millions of tiny switches that have to be either turned ON or OFF. It works only in two states: when the switch is in ON state its input is 1 and when the switch is in OFF state it is 0. These digits (0 and 1) are called bits and correspond to the OFF/ON positions of the switches in the computer processor. Thus, two digits match up to the ten digits to perform all mathematical operations. Therefore, to understand the language of computers, it is essential to study the binary number system. This chapter deals with the understanding of different number systems and their conversion from one base to another base. How to perform mathematical operations on binary numbers, representing signed binary numbers and performing binary arithmetic with signed numbers.

The numbers have to be represented in an easier manner to identify the number system to which they belong. This is usually done by representing numbers with a base. The base of any number system is defined as the number of digits in the system. For example, binary is a base-2 number system as it uses two digits, while decimal is a base-10 system as it uses 10 digits. The indication of base in a number helps us to carry out all operations easily. The base is represented as a small subscript after the number. Example: 537.15_{10}, 1010_{2}.

In general, it can be denoted as

$$\text{Digit } X_{\text{base position}}$$

DECIMAL NUMBER SYSTEM

Humans have been using decimal or base-10 numbering system extensively over years before the electronic age. This could be presumed as it relates to counting using 10 fingers. The decimal number system has a base of 10. This is probably the reason for assigning 10 for the base and not any other number. The decimal numbering system, also known as the denary system, is a positional system. The base-10 number system has 10 distinct digits (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). In decimal notation, we write a number as a sum of the individual digits multiplied by the power of 10 associated with the position of the digit. The digits in the rightmost end of the number is assigned ones position, the next digit to the left of it is the tens position, the next digit to the left is the hundreds position, the next digit to its left is thousands position and so forth. Therefore, any number is a sum of ones, tens, hundreds, thousands, and so on. Take the example of number 6231, it is represented as

$$6231 = (6 \times 1000) + (2 \times 100) + (3 \times 10) + (1 \times 1)$$

This can also be written using base 10 as $(6 \times 10^3) + (2 \times 10^2) + (3 \times 10^1) + (1 \times 10^0)$

The decimal system cannot be considered as the easiest system but due to the comprehensive usage, it can be considered the most intuitive and natural numbering system.

BINARY NUMBERING SYSTEM

The inventions of computers and other storage devices made the usage of denary system for counting illogical. When computers were designed to work with the decimal numbering system, it brought in a lot of complexity in terms of the circuitry, as the vacuum tubes kept increasing with the number to be represented. Hence, computers were incorporated with a program that was capable of translating decimal into binary at the input and later converting the binary back to decimal at the output. This disadvantage led to a much simpler concept of using binary digits. Owing to the fact, computers are built using transistors, and individual transistors possess only two options, ON or OFF states. In optical storage devices, data is stored depending on the light being reflected on the location or off the location. Likewise, magnetic storage devices store data at a specific location by magnetizing the particles in that location with a specific orientation. Hence, all of these devices have only two switching states, viz, ON and OFF. Therefore, it is essential to use two symbols that would be readily used by these devices. Such a numbering system that uses only two states is known as binary or base-2 system. The base-2 system has exactly two digits, 0 and 1. In a decimal system, the base-10 symbols are termed digits while in base-2 symbols are termed binary digits or bits. Data is stored as bits on computers, hard drives and for e-mail messaging.

OCTAL NUMBERING SYSTEM

The binary number system is very important as it is used with computations. Apart from the binary numbers there exists other important numbering systems that are frequently used. These are the octal and hexadecimal systems. Let us deal with octal

numbering first. The octal number system is a base-8 number system and uses the digits 0–7 to represent numbers. Binary to octal conversion is easier. Table 2.1 shows the octal conversion.

Table 2.1 Binary to octal conversion

8^4	8^3	8^2	8^1	8^0
4096	512	64	8	1

HEXADECIMAL NUMBERING SYSTEM

Binary numbers are often expressed as long chains of 0s and 1s. These strings of binary number sometimes confuse and was the biggest drawback of using the binary numbering system. Hence, to reduce the confusion, the hexadecimal or base-16, numbering system was used as an easier substitute to display the binary contents of storage devices. The bases of the binary system and hexadecimal system are multiples of two and hence a convenient link between these numbering systems can be obtained. In hexadecimal system, there are six more symbols to take us beyond 0–9. These are $A = 10$, $B = 11$, $C = 12$, $D = 13$, $E = 14$, $F = 15$. The numbering system equivalence shown in Table 2.2 illustrates that a single hexadecimal digit represents four binary digits. 8-bit or byte can be represented using two hexadecimal. The hexadecimal numbering system is used only for specific programs that analyse the binary display of memory. Hexadecimal numbering systems cannot operate computers but can be used to examine

Table 2.2 Numbering system equivalence table

Binary base-2	Octal base-8	Decimal base-10	Hexadecimal base-16
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	8	8
1001	9	9	9
1010	10	10	A
1011	11	11	B
1100	12	12	C
1101	13	13	D
1110	14	14	E
1111	15	15	F

the contents of RAM and to debug their programs. Thus, all number systems must be converted to either binary or decimal to achieve a better understanding of computers.

CONVERSION OF NUMBERS FROM ONE BASE INTO ANOTHER

Binary to Decimal

Binary numbers are positional with each position representing an increasing power of two starting with 2^0 for the rightmost digit. Each digit of the given binary number is multiplied, starting from the right.

Example 2.1 Convert $(11)_2$ to decimal equivalent.

Solution:
$$1 \times 2^1 + 1 \times 2^0 = 3_{10}$$

Example 2.2 Convert $(11101)_2$ to decimal.

Solution:
$$\begin{aligned} 11101 &= 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 16 + 8 + 4 + 0 + 1 = 29_{10} \end{aligned}$$

Table 2.3 Fractional powers of 2

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
0.5	0.25	0.125	0.0625	0.03125	0.015625

Binary system also consists of fractions apart from whole numbers. The simplest way of converting fractional binary numbers to decimal is by repeatedly multiplying the given number by fractional powers of two. Table 2.3 shows the fractional powers of 2. This is better understood using the following example.

Example 2.3 Convert 0.11 into decimal.

Solution:
$$\begin{aligned} 0.11_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 1 \times 0.5 + 1 \times 0.25 \\ &= 0.7510_{10} \end{aligned}$$

Example 2.4 Convert 0.1101 into decimal.

Solution:
$$\begin{aligned} 0.1101 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\ &= 1 \times 0.5 + 1 \times 0.125 + 1 \times 0.03125 \\ &= 0.65625 \end{aligned}$$

Decimal to Binary

A decimal number can be converted easily into an equivalent binary number by the division or remainder technique. The given decimal number is divided by 2 repeatedly and the remainder gives us the binary equivalent. Consider the following examples:

Example 2.5 Convert 21 into its binary equivalent.

Solution:

2	21	
2	10	R = 1
2	5	R = 0
2	2	R = 1
2	1	R = 0
	0	R = 1 ↑ $21_{10} = 10101_2$

The binary result reads the remainder. Therefore, the binary number is 10101.

Example 2.6 Convert 275 into its binary equivalent.

Solution:

2	275	
2	137	R = 1
2	68	R = 1
2	34	R = 0
2	17	R = 0
2	8	R = 1
2	4	R = 0
2	2	R = 0
2	1	R = 0
	0	R = 1

The binary number is $275_{10} = 100010011_2$

Decimal to Octal

Conversion of decimal numbers to octal numbers is obtained by dividing a given decimal number repeatedly by 8. The resulting remainders give the octal number.

Example 2.7 Convert 1654_{10} to octal form.

Solution:

8	1656	
8	207	R = 0
8	25	R = 7
8	3	R = 1
	0	R = 3

Answer: 3170
To cross verify = $3170 = 3 \times 512 + 1 \times 64 + 7 \times 8 + 0 = 1656$

Binary to Octal

The binary to octal conversion is much simpler. The given binary number is divided into groups of three starting from the least significant bit. The value of each group is given as the digit of the octal number.

Example 2.8 Convert 101010011_2 into octal form.

Solution:

101 010 011
5 2 3

Answer: $101010011_2 = 523_8$

Octal to Binary

The same technique employed previously is followed for converting octal to binary conversion. For every digit in the octal number, its 3-bit equivalent binary number is provided.

Example 2.9 Convert 7355_8 to binary form.

Solution:

7 3 5 5
111 011 101 101

Answer: $7355_8 = 111011101101_2$

Decimal to Hexadecimal

Conversion of decimal numbers to hexadecimal numbers follows the same division or remainder technique. Here, the decimal number is divided by 16 and the remainder gives the desired hexadecimal output. The remainder, in this case, is always one of the 16 hexadecimal digits.

Example 2.10 Convert 453 to hexadecimal.

Solution:

16	453	
16	28	R = 5
16	1	R = C
	0	R = 1

Answer: $453_{10} = 1C5_{16}$

Hexadecimal to Decimal

The same procedure followed for converting binary to decimal is used in converting hexadecimal to decimal. The hexadecimal number is multiplied by powers of 16 according to the position.

Example 2.11 Convert hex 3E7 into its decimal equivalent.

Solution:

$$= 3 \times 16^2 + E \times 16^1 + 7 \times 16^0$$
$$= 3 \times 256 + 14 \times 16 + 7 \times 1$$
$$= 768 + 224 + 7$$
$$= 999_{10}$$

Binary to Hexadecimal

The binary to hexadecimal is the simplest of all conversions. This is performed by grouping the given binary number in groups of four from left to right. They are replaced by its hexadecimal equivalent. The reverse process is used for converting hexadecimal to binary.

Example 2.12 Convert 11000101 to hexadecimal.

Solution:

1100 0101
C 5

Therefore, $(11000101)_2 = C5_{16}$

ARITHMETIC IN NUMBER SYSTEM

Binary Addition

Operations in binary arithmetic are similar to decimal arithmetic operations. The only difference is that the binary arithmetic involves only two digits: 0 and 1. The technique of addition for binary numbers is similar to that of decimal numbers, except that "1" is carried to the next column after both 1's are added.

The basic rules for binary addition is as shown in Table 2.4:

Table 2.4 Binary addition rules

Addend	0	0	1	1
Augend	+0	+1	+0	+1
Sum	0	1	1	0
Carry	0	0	0	1

These outputs can be implemented into the logic expressions. The logic expressions can then be designed as a logic circuit known as a "full-adder".

Example 2.13 Add the numbers 3_{10} and 1_{10} in binary form.

Solution: The decimal numbers are first converted to its binary equivalent. The numbers are 11 and 01. In the LSB column, 1 + 1 = 0, and has a carry of 1 to the next column. The next column adds up as 1 + 0 + 1 = 0 with a carry of 1 to the next column. The last column adds up as, 1 + 0 + 0 equals to 1.

$$\begin{array}{r} 11 \\ 01 \\ \hline 100 \end{array}$$

Thus, $11 + 01 = 100 = 4_{10}$.

Example 2.14 Add 7_{10} and 2_{10}.

Solution: To add the numbers $7 + 2$ in binary form, note that the binary form of 2_{10} is 010, while $7_{10} = 111$. The sum $7 + 2$ in binary form is shown as follows.

$$\begin{array}{r} 10 \\ 111 \\ \hline 1001 \end{array}$$

The right-hand column adds up as $1 + 0 = 1$. The next column adds up as $1 + 1 = 0$ with a carry of 1 to the next column. The leftmost column again is $1 + 1 = 0$ with a carry of 1 to the next column. Therefore, sum of binary numbers $10 + 111 = 1001 = 9_{10}$.

Example 2.15 Add 28 and 12 in binary form.

Solution:

$$\begin{aligned} 28_{10} &= 11100 \\ 12_{10} &= 1100 \\ &= 101000 \\ 101000 &= 2^5 + 2^3 \\ &= 32 + 8 = 40_{10}. \end{aligned}$$

The addition of fractional binary number is also similar. It has the following steps:

1. The given two numbers are aligned such that the radix points of each number are located in same column.
2. The blank places are filled with 0s, and then column-wise addition is performed.

Example 2.16 Add 110.01 and 1.011

Solution:

$$\begin{array}{r} 110.01 \\ +\quad 1.011 \\ \hline 110.010 \\ +\,001.011 \\ \hline 111.101 \end{array}$$

Hexadecimal Addition

Hexadecimal addition is also performed in the similar way. The individual assigned hexadecimal numbers are added. When the same hexadecimal digit is present in the

given numbers then this digit is taken as common and the decimal numbers are added. To this result, the hexadecimal digit is multiple. This is explained in the following example.

Example 2.17 Add 2H and AH.

Solution:

$$\begin{aligned} 2H + AH &= (2 + 10)\ H \\ &= 12H \\ &= CH \end{aligned}$$

BINARY SUBTRACTION

Binary subtraction can be performed using two ways. The easiest way is the borrowing subtraction. The second method is complement technique, which will be dealt with later. Let us consider the borrowing subtraction method or direct subtraction. It comprises of the following rules shown in Table 2.5.

Table 2.5 Binary subtraction rules

Rule 1	Rule 2	Rule 3	Rule 4
0	1	1	0
−0	−1	−0	−1
0	0	1	1

The first three rules are similar to the decimal subtraction but consider the fourth rule, here a bigger number is subtracted from a smaller number; therefore, to carry out the subtraction we need to borrow the next number which has a higher power. This is best understood from the following examples.

Example 2.18 Subtract 7_{10} from 9_{10}.

Solution: The numbers are converted to their binary equivalents.

$$\begin{aligned} 9 &= 1001 \\ -7 &= 0111 \\ \hline 2 &= 0010 \end{aligned}$$

1 in the first column is subtracted from 1 and gives 0. Since 1 in the second column cannot be subtracted from 0 we borrow from the next column. Since the third column also has only 0, 1 from last column having power 2^4 is borrowed. Therefore, the 1 remains and is hence the output.

Example 2.19 Subtract 1110_2 from 11101_2.

Solution:

$$\begin{aligned} &11011 \\ -&01110 \\ \hline &00101_2 \end{aligned}$$

NEGATIVE BINARY NUMBERS

Binary arithmetic deals with both positive and negative binary numbers. In decimal number systems, the negative number is represented with a negative sign before the number, e.g., –62 or –0.523 but this is not similar to negative binary numbers. Since computers deal only with 1's and 0's these negative binary numbers need to be modified. This problem of representing negative binary numbers is overcome by including an extra digit in front of the binary number, thus indicating it is positive or negative. The leftmost bit represents the sign and the remaining bits represent the magnitude. When a "0" is assigned in front of the number it indicates it is positive and when "1" is put in front of the number, it indicates it is negative. This digit or bit that stands for indicating the sign is called as sign bit. This approach is called the signed magnitude representation and is similar to assigning a negative sign in front of a number in decimal system. One disadvantage of this representation is that the number of bits must be mentioned; otherwise this may lead to confusion between the sign bit and the other bits.

Example 2.20 Convert decimal numbers -4_{10} to binary using signed magnitude representation.

Solution: Given decimal number is converted to binary number $4_{10} = 100_2$
Sign bit is added to it 0100
Negative number is represented by changing the sign bit 1100

The binary number should not be interpreted by including the first digit. This number is sometimes misinterpreted as 13 as they tend to include the first digit, however, since the first bit represents only sign, it should not be considered.

Answer: $-4_{10} = 1100_2$

These negative numbers can thus be represented in the sign magnitude representation in a simple way. It is to be noted that numbers with a sign can be easily computed and arithmetic operations are easier, whereas the negative binary numbers are quite complicated. Thus, we need to modify the way of indicating these numbers that enable simpler arithmetic operations. However, if these numbers are represented in a modified way for addition, then all computations should possess only addition circuitry which makes other operations likes subtraction and division impossible. To overcome this, we need to convert subtraction into an equivalent addition. Consider the decimal subtraction 5 – 2. It can be rewritten as 5 + (−2). Similarly, in binary system, all mathematical operations can be performed on negative numbers by using 1's complement and 2's complement.

One's Complement

Complementing a number is a technique where all the digits of the number are reversed. A signed binary number can be represented in 1's complement by changing

all the 0's to 1 and 1's to 0. While performing 1's complement, the following steps are to be noted.

For a given decimal number *n*

1. If *n* is positive, then it is simply converted to its binary equivalent.
2. If *n* is negative, then positive value of *n* in binary equivalent is written.
3. 1's are changed to 0 and 0's are changed to 1.
4. To the complemented number, 1is added.

Example 2.21 Represent -3_{10} in 1's complement method.

Solution:

$$3_{10} = 0011_2$$
$$-3_{10} = 1100_2$$

1's complement is used for subtraction of binary numbers. The given numbers are converted to their binary equivalent. The number that has to be subtracted is converted into 1's complement irrespective of whether it is smaller than the other number or not. The two numbers are then added together. During the process, if a carryover occurs while adding the most significant bits (MSB) of the two numbers, then the carry bit is added to the least significant bit of the result. The carry bit is often called as end around carry (EAC). The EAC occurs only for positive remainder and its absence indicates the remainder is negative and also in its complimented form.

Example 2.22 Subtract 8 from 10 using 1's complement method.

Solution:

10 = 1010
8 = 1000. The 1's complement is 0111
= 1010
+ 0111
10001
0001
+ 1
0010 = 2 (Answer)

Example 2.23 Subtract 10 from 8 using 1's complement method.

Solution:

10 = 1010
13 = 1101. The 1's complement is 0010
= 1010
+ 0010
1100 = −3

Since it does not have a carry, it is a negative number. Binary number and its equivalent decimal number in 1's complement notation is shown in Table 2.6. It is seen from the table that all negative decimal numbers start with 1 and the positive number starts with 0. The first bit however represents the sign bit as discussed earlier. The negative binary notations are complements of their equivalent positive notations.

Table 2.6 One's complement notation

Binary	Decimal	Binary	Decimal
0111	+7	1111	−0
0110	+6	1110	−1
0101	+5	1101	−2
0100	+4	1100	−3
0011	+3	1011	−4
0010	+2	1010	−5
0001	+1	1001	−6
0000	+0	1000	−7

Two's Complement

Two's complement is the second method of performing subtraction of negative binary numbers. Here, '1' is added to the 1's complement representation of the given negative binary number. Similar to 1's complement method, the 2's complement method also has the most significant bit indicating the sign of the number. The following steps are used to represent in the 2's complement method.

1. If the given number is positive, it is converted to its binary equivalent. If the given number is negative, the positive value of the number in binary is chosen.
2. Now the number is converted to its 1's complement form.
3. Finally, 1 is added to the 1's complement form.

Unlike the 1's complement representation, 0 in 2's complement method has only one way of representing 0 and are retained. This is a major advantage with the 2's complement method. In 2's complement subtraction, if any carry bit occurs, it is simply discarded whereas in 1's complement, if the carry occurs, then the extra bit is added again to the remainder that results in the correct answer. In 2's complement if carry is present, the resulting number is considered as positive number and when there is no carry, it is considered as negative number. However, with 2's complement, we ignore this extra bit. No other computations are required to find the correct answer. This makes 2's complement a more efficient way of representing signed numbers and performing binary subtraction. Table 2.7 represents the 2's notation of decimal.

Table 2.7 Two's complement notation

Binary	Decimal
0111	+7
0110	+6
0101	+5
0100	+4
0011	+3
0010	+2
0001	+1
0000	+0
1111	−1
1110	−2
1101	−3

Example 2.24 Represent –2 in 2's complement.

Solution:

$$\begin{aligned} 2 &= 0010 \\ &\quad 1101 - \text{1's complement} \\ &+ \quad 1 - \text{results in 2's complement} \\ &\quad 1110 = -2 \end{aligned}$$

Example 2.25 Subtract 2 from 7 using 2's complement method.

Solution: This can be done by adding –2 to 7.

$$\begin{aligned} 2 &= 0010 \\ -2 &= 1101 \\ + \quad & \quad 1 \\ \hline & \quad 1110 \\ 7 &= 0111 \\ \hline & \quad 10101 \end{aligned}$$

The MSB is discarded, therefore, the answer = 0101 = 5

BINARY MULTIPLICATION

Binary multiplication is similar to decimal multiplication. Binary multiplication is much easier compared to the other multiplication forms as only two digits 0 or 1 is used for multiplication. While performing binary multiplication, the following rules are to be considered:

1. Multiplication of the given number with 1 gives the same number and when multiplied with 0, the product is 0.

2. The results are moved one column to the left for the successive products, similar to the procedure followed during decimal multiplication.
3. The resultant binary numbers are added to get the final product.

Example 2.26 Multiply 101 with 100.

Solution:

$$\begin{array}{r} 101 \\ \times\ 100 \\ \hline 000 \\ 000x \\ 101xx \\ \hline 10100 \end{array}$$

Example 2.27 Multiply 1010 with 1010.

Solution:

$$\begin{array}{r} 1010 \\ \times\ 1010 \\ \hline 0000 \\ 1010x \\ 0000xx \\ 1010xxx \\ \hline 1100100 \end{array}$$

BINARY DIVISION

Binary division can be performed by the technique of long division in decimal division. Binary division however uses only 1's and 0's. The following rules are to be taken care while performing binary division:

1. While performing division, if the divisor is less than or equal to the remainder, then 1 is added to the quotient and subtracted.
2. When the divisor is greater than the remainder, 0 is written in the quotient and to this the next digit from the dividend is added.

When the division is carried out for all the digits of the dividend and if a remainder still exists, then a radix point is marked in the dividend and a zero is assigned. Fractions represented in binary have different representation than those in decimal system and hence, not all binary numbers terminate on division.

Example 2.28 Consider the decimal division of 23 by 5.

Solution:

$$\begin{array}{r} 5\overline{)23}(4 \\ \underline{20} \\ 3 \end{array}$$

The divisor is 5, dividend is 23 and the remainder is 4. The same procedure is followed in binary division also.

Example 2.29 Divide 111 by 10.

Solution: The divisor is 10 and dividend is 111.

```
10)111(11.1
   10
   11
   10
    10
    10
     0
```

Answer: 11.1

Example 2.30 Divide 1100011_2 by 110_2.

Solution:

```
110)11011(100.1
   110
   001
   000
   011
   000
   110
   110
     0
```

NUMBER CODES

The numbers or letters are represented as groups and are assigned specified codes in form of symbols that facilitate easy counting. The process of assigning codes to numbers or words/letters is called encoding. A single symbol or a group of symbols representing a set of numbers is known as a code. Since in digital computing the data is stored in the form of group of bits, this method of coding is referred to as binary code. The binary codes are classified into many sub categories. They are as follows:

- Weighted codes
- Non-weighted codes
- Binary coded decimal code
- Alphanumeric codes
- Error detecting codes
- Error correcting codes

Weighted Codes

Weighted binary codes are those binary codes which have a specific weight for each position of the given number. Each decimal digit of the given number is represented as a group of 4-bit number. Several systems of the codes are used to express the decimal digits 0 through 9. The weighted codes are 8421 code, 84-2-1 code, 2421 code, and 5211 code. The negative sign is represented as a bar on the number.

Binary Coded Decimal (BCD) or 8421 Code

The binary coded decimal code, abbreviated as BCD, is a method that uses binary digits "0" and "1". ON state represents "1" and OFF state represents "0". Each digit is called a bit. 4-bits are called a nibble and is used to represent each decimal digit (0 through 9). This coding system has been used since the first computer. This coding system deals with decimal and binary numbers. Each decimal number requires 4 bits to code them. Example $(51)_{10}$ is represented as 0101 0001 using BCD code. The number of bits required to code binary numbers is more when compared to binary coding. Despite the fact that more bits are used to represent a decimal number in BCD code still it is convenient to use for input and output of operations in digital systems. The code is also known as 8-4-2-1 code. This is because 8, 4, 2, and 1 are the weights of the 4 bits of the BCD code. The 8, 4, 2, 1 code is obtained by taking the first 10 binary numbers and assigning them to the corresponding decimal digits. The number 243 would be represented as 0010 0100 0011 in the 8, 4, 2, 1 code. There are 17 different sets of weights possible for a positively weighted code: (3, 3, 3, 1), (4, 2, 2, 1), (4, 3, 1, 1), (5, 2, 1, 1), (4, 3, 2, 1), (4, 4, 2, 1), (5, 2, 2, 1), (5, 3, 1, 1), (5, 3, 2, 1), (5, 4, 2, 1), (6, 2, 2, 1), (6, 3, 1, 1), (6, 3, 2, 1), (6, 4, 2, 1), (7, 3, 2, 1), (7, 4, 2, 1), (8, 4, 2, 1).The arithmetic operations are restricted to only few of these codes since these arithmetic operations are very difficult in almost all of the possible codes. The maximum decimal digit available is 9_{10}. The binary codes for numbers from 10 to 15 are therefore not used in BCD code. The BCD codes for six decimal numbers from 10 to 15 are given in Table 2.8 below.

Table 2.8 Decimal numbers and its equivalent BCD codes

Decimal number	BCD code
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

These six codes are called forbidden codes and the group of these codes is called the forbidden group in BCD code. Weights of the numbers can be represented as negative representatives as in 8, 4, −2, −1 code. The complements of the numbers are taken for negative representation. Thus, this is self-complementing code. Each bit is

individually complemented (by changing 1's to 0's and 0's to 1's) and this new code word represents the complement of the digit to which the original code word corresponds. For example, 0101 represents 3 in the 8, 4, –2, –1 code, and 1010 represents 6 in this code. It is a weighted and self-complementing code. The different weighted codes for decimal digits 0 to 9 is shown in Table 2.9.

Table 2.9 The different weighted codes for decimal digits 0–9

Decimal code	BCD or 8 4 2 1	84–2–1	7421	5421	5311	4221	2421	Biquinary 5043210
0	0000	0000	0000	0000	0000	0000	0000	0100001
1	0001	0111	0001	0001	0001	0001	0001	0100010
2	0010	0110	0010	0010	0011	0010	0010	0100100
3	0011	0101	0011	0011	0100	0011	0011	0101000
4	0100	0100	0100	0100	0101	1000	0100	0110000
5	0101	1011	0101	0101	1000	0111	1011	1000001
6	0110	1010	0110	0110	1001	1100	1100	1000010
7	0111	1001	1000	0111	1011	1101	1101	1000100
8	1000	1000	1001	1011	1100	1110	1110	1001000
9	1001	1111	1010	1100	1101	1111	1111	1010000

Advantages of BCD Codes

- BCD coding is similar to the binary equivalent of decimal numbers 0–9.
- BCD has no limitation for number size.
- It is easier to convert decimal numbers from or to BCD than to binary form.

Disadvantages of BCD Codes

- Each decimal number requires four bits to be represented in the BCD code.
- Arithmetic operations in BCD or weighted codes are much complicated as it deals with more number of bits and also it has different set of rules.
- BCD is less efficient than binary.

Excess-3 Codes

The excess-3 code is obtained by adding $(0011)_2$ or $(3)_{10}$ to the desired 8421 code. A given decimal number is converted to its excess-3 code by adding "3" to each decimal digit in the given number and replacing each digit of the newly found decimal number by its 4-bit binary equivalent. The addition of "3" to a binary digit in 8421 coded form produces a carry and is not taken forward. The result of addition should be taken as a single entity and subsequently replaced with its excess-3 code equivalent.

The excess-3 code is also called XS-3 code. It is non-weighted code used to express decimal numbers. The difficulties encountered during arithmetic operations are overcome by using excess-3 codes. This makes the excess-3 codes significant for arithmetic operations. The excess-3 code for decimal numbers 0–9 are listed in the following Table 2.10.

Table 2.10 Excess codes

Decimal number	Excess-3 code
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100
10	0100 0011
11	0011 0011

Example 2.31 Find the Excess-3 code for the following:

(i) 24

(ii) $(156.75)_{10}$

(iii) 1000 1100 1010

Solution:

(i) The decimal number 24 is represented in BCD code with the individual binary equivalents. To this binary number, 0011 is added which results in the excess-3 code for 24_{10}.

$$\begin{array}{cc} 2 & 4 \\ 0010 & 0100 \end{array}$$

Answer: 0101 011

The excess-3 code can also be obtained by directly adding 3 to 2 and 4. This adds up to 5 and 7. It is written in the binary code.

(ii) $(151.75)_{10}$

The integer part is first converted into excess binary code. The excess-3 code for 151 is 0100 1011 0100. The fractional part .75 is converted by replacing 7 and 5

with the 4-bit binary equivalents of 10 and 8, respectively. The excess-3 code for $(.75)_{10}$ is (.10101000).

Answer: $(151.75)_{10}$ in Excess-3 code is 010010110100.10101000.

(iii) The excess-3 code for the given binary number is obtained by subtracting 0011 from each 4-bit group.

1000 1100 1010
5 9 7

The binary number thus obtained after subtraction is 0011 0100 01 Excess-3 code is 010110010111 and its binary equivalent is 264_{10}.

GRAY CODE

In 1953, Frank Gray proposed and patented the Gray code. A given word length can possess more than one Gray code but this Gray coding was initially restricted only to denote non-negative integers called the binary-reflected Gray code. Unlike the other coding systems, the Gray code is not an arithmetic code, and it is not used in circuits dealing with arithmetic operations but is used for input and output operations in digital systems. It does not have any specific weight for the position of a bit and hence is a non-weighted code. The Gray code has a distinctive trait, for every bit added to the decimal number only one bit changes at a particular time. After each successive increase, the resultant number differs by 1-bit. Since only 1-bit changes at a time, the Gray code is called a unit distance code. The number of bits required to represent a number in Gray code is less when compared to binary codes which reduces errors caused during encoding data to a maximum extent. The binary and Gray code equivalents of decimal numbers 0–15 is listed in Table 2.11.

From the table it is evident that the consecutive numbers vary only by 1-bit. This is known as the cyclic property of the Gray code.

Conversion of Binary to Gray Code

The following steps are used for converting a binary number to its Gray code equivalent.

1. The most significant bit (MSB) of the given binary number is retained. This MSB is the same for the Gray coded number also.
2. The bit adjacent to the MSB which is the second most significant bit is now obtained by adding the first MSB and the second MSB of the binary number. If carry occurs, it is neglected.
3. The next significant bit of the Gray code is obtained by adding the second MSB and the third MSB in the binary number. Carry is ignored.
4. The method is continued until the LSB of the Gray code number is obtained by adding the LSB and the next higher adjacent bit of the binary number.

Table 2.11 4-bit gray code

Decimal code	Gray code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

Example 2.32 Convert $(1010)_2$ to Gray code.

Solution:

Step 1: 1010 – Binary
1 – Gray code (MSB is retained)

Step 2: 1010
11 (MSB is added to the next bit)

Step 3: 1010
111 (Second MSB is added to the next bit)

Step 4: 1010
1111 (Third MSB is added to the LSB)

Gray code = 1111

Conversion of Gray Code to Binary

The following steps are used for converting a binary number to its Gray code equivalent.

1. The most significant bit (MSB) of the given Gray coded number is retained. This MSB is the same for the binary number also.
2. The bit adjacent to the MSB which is the second most significant bit in now obtained by adding the MSB and the second MSB of the Gray number. If carry occurs, it is neglected.

3. The next significant bit of the binary number is obtained by adding the second MSB and the third MSB in the Gray number. Carry is ignored.
4. The method is continued until the LSB of the binary number is obtained by adding the LSB and the next higher adjacent bit of the gray coded number.

Example 2.33 Convert 1101 to Binary code.

Solution:

Step 1: 1101 – Gray code
1 – Binary code (MSB is retained)

Step 2: 1101
10 (MSB is added to the next bit, carry exist since binary addition 1+1 = 10, therefore, 0 is retained and 1 is ignored)

Step 3: 1101
100 (Second MSB is added to the next bit)

Step 4: 1101
1001 (Third MSB is added to the LSB)

Binary code = 1001_2

Non-Boolean Gray Code

Gray code is of two types, the one discussed above is the binary-reflected Gray code. The second type is the non-Boolean Gray code. The non-Boolean Gray code uses non-Boolean symbols for encoding. The generalized representation of the code is the (n, k)-Gray code, where n is the number of independent digits used and k is the word length. When n is odd, then Gray code does not have the cyclic property and when n is even it possess the cyclic property.

Advantages of the Gray Code

- Only one bit changes for every addition.
- Number of bits involved is minimized as compared to the other coding forms.
- Errors during encoding can be minimized.

Limitations of the Gray Code

- It is not used in arithmetic circuits.

Applications of the Gray Code

- The Gray code is used in the input and output digital signals.
- The Gray code is preferred in shaft angle measuring devices as it eliminates errors during the process.

- Karnaugh mapping, a technique used for minimization of Boolean expressions, uses Gray code for labelling the axes of the maps.
- Power consumption is less and hence used in memory in computers.
- Used in algorithms to detect incremental changes.

ALPHANUMERIC CODES

Communication between computers deals with alphabets and symbols apart from numerals. Since more symbols are required for communication, the alphanumeric codes were introduced. These included alphabets in capital and small letters, punctuation marks, and symbols. These codes are not number codes but character codes that are used for giving instructions or commands to computers or any digital device. These codes are also used during interfacing computers with other devices like printers, scanners, etc. Alphanumeric codes are of different types.

Hollerith Code

Herman Hollerith in 1890 invented the Hollerith codes that were represented as punches in cards. They were first used tabulated census data. These cards were later used to store and encode alphanumeric data. However, the code became outdated on the invention of computers. These cards could not withstand physical damage since folding, stapling or multilating of cards were restricted.

American Standard Code for Information Interchange (ASCII)

The ASCII (American Standard Code for Information Interchange), pronounced "ask-ee" is a 7-bit alphanumeric code used in computers and electronic devices. The code was standardized in 1967 and later updated and published as ANSI X3.4-1968, ANSI X3.4-1977, and finally as ANSI X3.4-1986. This code is used in microcomputers, PCs, and workstations. The ASCII code can represent only 128 characters since it is a 7-bit code. It has 95 printable characters including 26 upper-case letters (A to Z), 26 lower-case letters (a to z), 10 numerals (0 to 9), and 33 special characters that include mathematical operators, punctuation marks, and space character. Apart from these it comprises of 33 non-printing characters which are outdated. The version of ASCII code is denoted as the number beside the code example US ASCII-8 or ASCII-8. The ASCII code is generally a 7-bit code with the 8-bit which is the MSB used for parity check. The ASCII code for first few decimal numbers is denoted in the following Table 2.12.

ASCII code has been developed over the years to improve its compatibility with other languages. Although it has a few limitations, it still is the most common character encodings used till date.

Table 2.12 ASCII code

Decimal	Hex	Binary	Code	Code description
0	00	0000 0000	NUL	Null character
1	01	0000 0001	SOH	Start of header
2	02	0000 0010	STX	Start of text
3	03	0000 0011	ETX	End of text
4	04	0000 0100	EOT	End of transmission
5	05	0000 0101	ENQ	Enquiry
6	06	0000 0110	ACK	Acknowledgement
7	07	0000 0111	BEL	Bell
8	08	0000 1000	BS	Backspace
9	09	0000 1001	HT	Horizontal tab
10	0A	0000 1010	LF	Line feed
11	0B	0000 1011	VT	Vertical tab
12	0C	0000 1100	FF	Form feed
13	0D	0000 1101	CR	Carriage return
14	0E	0000 1110	SO	Shift out
15	0F	0000 1111	SI	Shift in
16	10	0001 0000	DLE	Data link escape
17	11	0001 0001	DC1	Device control 1 (XON)
18	12	0001 0010	DC2	Device control 2
19	13	0001 0011	DC3	Device control 3 (XOFF)
20	14	0001 0100	DC4	Device control 4

Applications

- They are mainly used to give instructions to microcomputers
- It is used for interfacing devices
- It is used in the transfer of information from keyboards to computer displays and printers

Extended Binary Coded Decimal Interchange Code (EBCDIC)

The second commonly used alphanumeric code is EBCDIC (Extended Binary Coded Decimal Interchange Code), pronounced as “eb-si-dik”. This is an 8-bit code developed by IBM mainly used in larger computers. EBCDIC code finds its application in all IBM mainframe computer peripherals and operating systems. A single byte in EBCDIC is divided into two 4-bit groups called nibbles. The first 4-bit group

represents the category of the character and is called the "zone", while the second 4-bit group identifies the specific character and is called the "digit". EBCDIC can correspond to 256 characters. EBCDIC does not provide any advantage over ASCII code except that it is easier to enter data into larger machines. The EBCDIC code for first few decimal number is provided in Table 2.13.

Table 2.13 EBCDIC code

Decimal	Hex	Binary	Code	Code description
0	00	0000 0000	NUL	Null character
1	01	0000 0001	SOH	Start of header
2	02	0000 0010	STX	Start of text
3	03	0000 0011	ETX	End of text
4	04	0000 0100	PF	Punch off
5	05	0000 0101	HT	Horizontal tab
6	06	0000 0110	LC	Lower case
7	07	0000 0111	DEL	Delete
8	08	0000 1000		
9	09	0000 1001		
10	0A	0000 1010	SMM	Start of manual message
11	0B	0000 1011	VT	Vertical tab
12	0C	0000 1100	FF	Form feed
13	0D	0000 1101	CR	Carriage return
14	0E	0000 1110	SO	Shift out
15	0F	0000 1111	SI	Shift in
16	10	0001 0000	DLE	Data link escape
17	11	0001 0001	DC1	Device control 1
18	12	0001 0010	DC2	Device control 2
19	13	0001 0011	TM	Tape mark
20	14	0001 0100	RES	Restore
21	15	0001 0101	NL	New line
22	16	0001 0110	BS	Backspace
23	17	0001 0111	IL	Idle
24	18	0001 1000	CAN	Cancel
25	19	0001 1001	EM	End of medium
26	1A	0001 1010	CC	Cursor control
27	1B	0001 1011	CU1	Customer use 1
28	1C	0001 1100	IFS	Interchange file separator
29	1D	0001 1101	IGS	Interchange group separator

(Continued)

Decimal	Hex	Binary	Code	Code description
30	1E	0001 1110	IRS	Interchange record separator
31	1F	0001 1111	IUS	Interchange unit separator
32	20	0010 0000	DS	Digit select
33	21	0010 0001	SOS	Start of significance
34	22	0010 0010	FS	Field separator
35	23	0010 0011		
36	24	0010 0100	BYP	Bypass
37	25	0010 0101	LF	Line feed
38	26	0010 0110	ETB	End of transmission block
39	27	0010 0111	ESC	Escape
40	28	0010 1000		
41	29	0010 1001		
42	2A	0010 1010	SM	Set mode
43	2B	0010 1011	CU2	Customer use 2
44	2C	0010 1100		
45	2D	0010 1101	ENQ	Enquiry
46	2E	0010 1110	ACK	Acknowledge
47	2F	0010 1111	BEL	Bell
48	30	0011 0000		

PARITY METHOD FOR ERROR DETECTION

Error is said to be any unwanted change in the output information as compared to the input information. In Analog systems, there is a possibility of some undesired signals being received at the receiver. These undesired signals are referred as noise. This causes error during signal transmission. On the other hand, in digital systems since the information is transferred as bits, errors occur due to interchanging of bits.

Error is produced when a bit changes from “0” to “1” or “1” to “0” while data is transferred from one computer to another. The process of error occurrence is as shown in Figure 2.1. The error caused might even corrupt the data transferred. Therefore, it is essential to check if any error is caused during the process. Methods have to be evolved

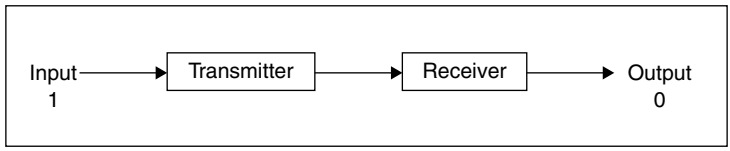


Fig.2.1 Error occurrence

to keep the error in check so that once an error is detected it has to be notified. To overcome this problem, digital devices use the error detecting code and its correction. The error detection code is added to the input data and if there is an error that had occurred during the transmission of the data then the code detects and helps us trace the error.

Apart from the error detection codes, there are certain other codes that can be sent along with the data which not only perform the operation of the detecting the error but also help us locate the exact bit position of the error occurrence. It identifies the bit that has undergone transformation. Such codes are known as the error-correcting code. The error correction codes after detecting the particular incorrect bit does the necessary change so that its value is reverted to the original data.

The simplest method employed for the detection and correction of these errors is the parity check. Parity check involves the process of adding one or more than one extra bits to the input data bits during the transmission. The additional bits added are called parity bits and perform detection or correction of the errors. The code word is formed by combining the data bits along with the parity bits.

Consider an 8-bit word. The MSB is the parity bit and the remaining 7-bit are used as data bits. The parity of the transmitted word can either be even parity or odd. If the total number of 1s in the transmitted word is an odd number after adding the parity bit then the transmitted word has an odd parity and if the total number of 1s in the transmitted word is an even number after adding the parity bit then the transmitted word has an even parity. The parity bit is always placed on the left of the most significant bit. Examples for odd and even parity are shown in the following Table 2.14.

Table 2.14 Examples for even and odd parity

Parity bit	Data bit	Total number of 1s	Parity
1	1011001	5	Odd
0	1011001	4	Even
1	1000001	3	Odd
0	1000001	2	Even
1	1101011	6	Even
0	1101011	5	Odd
1	0000000	1	Odd
0	0101010	3	Odd
1	1111111	8	Even
0	1111111	7	Odd

It is evident from Table 2.14 that for the same data bit the parity can be made even or odd depending on the parity bit. Thus, to obtain an even parity the parity bit is set to 1 or 0 such that the total number of "1" in the entire word is even and for an odd parity, the parity bit can be set to 0 or 1 depending such that the number of "1-bit" in the entire word is odd. The process involved in correcting an error is depicted in Figure 2.2.

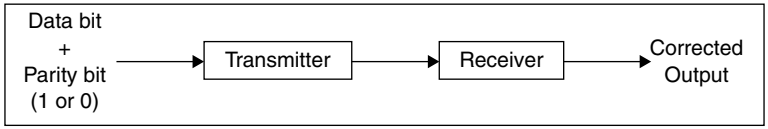


Fig.2.2 Corrected output using parity bit

Thus, the data can be corrected by setting the parity bit. The error is identified at the receiver by noting if there exists any change in the parity other than the expected data. If the receiver signal or data is expected to be of odd parity but an even parity occurs then we can confirm for an error. The receiver does not take into account this byte and retransmits the same to the transmitter.

Repetition Code

Another useful method used in error detection is the repetition code. The presence of error is displayed as repeated bits in the output. Consider for example, if the received data has a two-fold repetition of "1" and "0" as "11" and "00", respectively, then the occurrence of an error can be detected. There is possibility for detection of single bit errors also since that particular bit occurs repeatedly. Instead of the input data "01" the output occurs as 011 or 101 or 110 and similarly, it can also be transmitted as 001 or 010 or 100. In both cases, the bit that has occurred repeatedly is considered as the correct bit. If an 8-bit data is to be transmitted then considering the entire bit at once is difficult hence they are sent in blocks or parts. The first 4-bit is transmitted first and checked for repetitions. But this code proves to be inefficient when the number of repetitions increases as the bit involved is more. If the repetition of bit increases from three-fold to five-fold then accommodating 2-bit data becomes difficult. Thus, the repetition code is highly inefficient as the number of times each bit is repeated has to be increased to obtain a standardized error detection and correction.

Example 2.34 Write the even parity code and three-fold repetition code for:

1. 0001
2. 1111
3. 1010
4. 1100
5 0000

Solution:

1. The parity code for 0001 is 10001 and three-fold repetition code is 000100010001.
2. The parity code for 1111 is 01111 and three-fold repetition code is 111111111111.
3. The parity code for 1010 is 01010 and three-fold repetition code is 101010101010.
4. The parity code for 1100 is 01100 and three-fold repetition code is 101010101010.
5. The parity code for 0000 is 00000 and three-fold repetition code is 000000000000.

Cyclic Redundancy Check Code

The cyclic redundancy check codes (CRC) provides a reasonably high level of protection even at low redundancy levels. To generate the CRC codes two methods are adopted.

1. Arithmetic division by modulo 2
2. Polynomial division method

In arithmetic division method the data bit sequence is divided by modulus 2. The resultant remainder obtained as a result of mod-2 division is added to the dividend bit sequence. The generated code is completely divisible by the divisor used in the generation of the code. The second method to obtain the cyclic redundancy check code is the polynomial division method.

The polynomial division on the other hand is also of equal importance. An algorithm similar to the standard integer division is for the polynomial division. While performing a polynomial division the remainder should be a polynomial of lesser degree than the divisor. This is taken into consideration while generating the cyclic redundancy check code. Consider a generator polynomial G(x) whose coefficients are all either 0 or 1.

$$G(x) = x^3 + x^2 + 1 \tag{2.1}$$

If the transmitted data in the form of bits are represented as $b_n, b_{n-1}, b_{n-2}, \ldots b_2, b_1, b_0$, these bits are taken as the coefficients of the polynomial such that

$$B(x) = b_n x^n + b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + \ldots b_2 x^2 + b_1 x + b_0 \tag{2.2}$$

This polynomial is multiplied by x^k where k is the degree of the generator polynomial. The resulting product is divided by the generator to obtain two other polynomials Q(x) and R(x) such that

$$x^k B(x) = Q(x)\,G(x) + R(x) \tag{2.3}$$

All coefficients of the polynomial are considered not as integers but as integers of modulo 2. The coefficients of the remainder polynomial are considered as parity bits. If the degree of the R(x) is less than k then the transmitted message is given as:

$$x^k B(x) + R(x) \tag{2.4}$$

This polynomial remains the same for both addition and subtraction. Equation 2.4 can be simplified as

$$x^k B(x) - R(x) = Q(x)\,G(x) \tag{2.5}$$

Thus, from the above equation it is clear that the coefficient of the given polynomial is considered as the transmitted bit and therefore polynomial will be divisible by G(x). This forms the basis of error checking using the CRC. When a message is received the corresponding polynomial is divided by G(x). If the remainder is non-zero then an error has occurred and it is detected. When the remainder is zero then the transmitted message does not have error occurrence.

Let us consider an input data 00000111 that has to be checked for error using CRC. The input data corresponds to the polynomial

$$x^3 + x^2 + 1$$

The generator polynomial $G(x) = x^3 + x^2 + 1$ and has a degree '3'. Therefore, the given polynomial is multiplied using x^3.

$$x^6 + x^5 + x^3$$

The resultant polynomial is divided by G(x).

$$\begin{array}{r|l} & x^3 \\ \hline x^3 + x^2 + 1 & x^6 + x^5 + x^3 \\ & x^6 + x^5 + x^3 \\ \hline & \end{array}$$

The Remainder is 0. Therefore, no error has occurred.

Let us consider an input data 1101001 that has to be checked for error using CRC. The input data corresponds to the polynomial

$$x^3 + x^2 + 1$$

The generator polynomial $G(x) = x^3 + x^2 + 1$ and has a degree '3'. Therefore, the given polynomial is multiplied using x^3.

$$x^7 + x^6 + x^4 + 1$$

The resultant polynomial is divided by G(x).

$$x^{10} + x^9 + x^7 + x^3$$

The resultant polynomial is divided by G(x).

$$\begin{array}{r|l} & x^7 + 1 \\ \hline x^3 + x^2 + 1 & x^{10} + x^9 + x^7 + x^3 \\ & x^{10} + x^9 + x^7 \\ \hline & x^3 \\ & x^3 + x^2 + 1 \\ \hline & -x^2 - 1 \end{array}$$

The remainder is 1, therefore the parity bits added in this case would be 001. Thus, generated code word is completely divisible by the divisor used in the generation of the code. An error-free reception leading to an all zero remainder is possible when the received code word is again divided by the same divisor. A non-zero remainder is indicative of the presence of errors. The number of check bits, *n*, used in the construction of the cyclic code determines the probability of error detection. The probability of error detection is 100% for:

- Single-bit errors
- Two-bit errors
- When an error is present in odd number of bits
- When the error burst has a length less than $n + 1$

But the probability of error detection reduces to $1-\left[\frac{1}{2}(n-1)\right]$ for an error burst length $n+1$ and to $1-\left(\frac{1}{2}\right)n$ for an error burst length greater than $n+1$.

Hamming Code

The error detection and correction codes described in cyclic redundancy check explains how the capability of code detection and error correction can be improved by increasing the number of redundant bits added to message bits. This method was introduced by RW Hamming. By increasing the number of parity bits, the error apart from being detected and located can also be corrected. The addition of the bits brings changes in the Hamming distance code parameter. The dissimilarity between the number of bits in two-coded words is referred to as the Hamming distance. On the addition of a single-bit parity to a code, the Hamming distance is 2. A three-fold repetition code has a Hamming distance of 3. Thus, it was noticed that by increasing the Hamming distance, the error detection and correction can be improved.

The generalized form for the Hamming code is as follows:

- The parity bits were always placed at positions having powers of 2 in a code.
- The remaining positions were represented by data bits.
- Therefore, the generalized form of the code is $P_1P_2D_1P_3D_2D_3D_4P_4D_5D_6D_7D_8D_9D_{10}D_{11}P_5$ _ _ _, where P and D represent parity and data bits respectively.
- The parity bit takes values either 0 or 1 depending on the desired parity. The parity bits are formed as groups.
- These groups are formed by examining the $N-1$ bits and later followed by checking N-bits following the parity bit, where N is the position of the parity bit. For the parity bit P_1 the position $N=1$, for P_2 the value of $N=2$, $N=4$ for P_3, and so on.

The Hamming code is capable of correcting single-bit errors on messages of any length. One of the most repeatedly used Hamming code is the 7-4 code that uses a coded word comprising of 7-bit, four data bits, and three parity bits. This is represented as $P_1P_2D_1P_3D_2D_3D_4$. Here P_1, P_2, and P_3 are the parity bits and D_1, D_2, D_3, and D_4 are the data bits. The process of writing the Hamming code and thereby the detection and identification of error bits is explained using the example. The Hamming code is written such that for the first parity bit P_1 an even parity is created in the positions 1, 3, 5, 7 of the word. P_2 is chosen such that an even parity is created in positions 2, 3, 6, and 7. The third parity P_3 should produce an even parity in the positions 4, 5, 6, and 7 of the word. This is best understood with an example.

Consider the Hamming code for four bit 1001. The bits present in positions 1, 3, 5, 7 of the 7-bit data are P_1, 1, 0, 1, to establish an even parity P_1 must be 0. Similarly, the bits in the position 2, 3, 6, and 7 are P_2, 1, 0, and 1. To create an even parity, the value of P_2 must be 0. Lastly, the bits present at positions 4, 5, 6, 7 is P_3, 0, 0, 1, therefore, the value that has to be assigned to P_3 in order to achieve even parity is 1. Table 2.15 explains the values of the parity bits and hence the resultant code.

Table 2.15 Generation of hamming code for the word 0110

Status	P_1	P_2	D_1	P_3	D_2	D_3	D_4
Position	1	2	3	4	5	6	7
Word without parity bits			1		0	0	1
Addition of parity bit P_1	0		1		0		1
Addition of parity bit P_2		0	1			0	1
Addition of parity bit P_3				1	0	0	1
Word with parity bits	0	0	1	1	0	0	1

Thus, the Hamming code for 0110 is 0011001.

Let us assume that the data 0011001 is transmitted and data bit D_2 is changed while transmission. The code received is 0001101. The error is detected by checking the parity for three combinations mentioned above, namely, $P_1D_1D_2D_4$, $P_2D_1D_3D_4$ and $P_3D_2D_3D_4$. The process results in displaying three additional parity bits at the receiving end of the transmission. The parity for the group $P_1D_1D_2D_4$ is assumed as X, $P_2D_1D_3D_4$ is Y and $P_3D_2D_3D_4$ is Z. If these bits are "0" then the parity is unchanged and there is no error in the transmitted data. But when these bits are displayed as "1" then there exist error in the transmitted data. When the parity is checked for the first group and results in $X = 1$ since it does not have even parity. The second group has an even parity thus $Y = 0$. The third relation does not has an even parity and hence $Z = 1$. Thus, the resultant parity is written in reverse order, i.e., ZYX which gives the position of the bit in which error has occurred and has to be corrected. Thus, the bit in which error has taken place is 101 whose binary equivalent is "5". Thus, the fifth bit D_2 has to be corrected. Once on correcting the bit in position 5, the corrected code is received.

Example 2.35 Write the even parity Hamming code for:

1. 1010 2. 1100

Solution:

1. The generalized form of the Hamming code $P_1P_2D_1P_3D_2D_3D_4$ is filled with the given data bits. This results in $P_1P_21P_3010$. The three groups $P_1D_1D_2D_4$, $P_2D_1D_3D_4$, and $P_3D_2D_3D_4$ are checked for the parity as shown in Table 2.16.

Table 2.16

Status	P_1	P_2	D_1	P_3	D_2	D_3	D_4
Position	1	2	3	4	5	6	7
Word without parity bits			1		0	1	0
Addition of parity bit P_1	1		1		0		0
Addition of parity bit P_2		0	1			1	0
Addition of parity bit P_3				1	0	1	0
Word with parity bits	1	0	1	1	0	1	0

Therefore, the transmitted Hamming code is 1011010.

2. The generalized form of the Hamming code $P_1P_2D_1P_3D_2D_3D_4$ is filled with the given data bits. This results in $P_1P_21P_3010$. The three groups $P_1D_1D_2D_4$, $P_2D_1D_3D_4$, and $P_3D_2D_3D_4$ are checked for the parity as shown in Table 2.17.

Table 2.17

Status	P_1	P_2	D_1	P_3	D_2	D_3	D_4
Position	1	2	3	4	5	6	7
Word without parity bits			1		1	0	0
Addition of parity bit P_1	0		1		1		0
Addition of parity bit P_2		1	1			0	0
Addition of parity bit P_3				1	1	0	0
Word with parity bits	0	1	1	1	1	0	0

Therefore, the transmitted Hamming code is 0111100.

Example 2.36 A 7-bit Hamming code is received as 1001001. Find the error position and correct the code.

Solution: The received Hamming code is 1001001. The parity is checked for three groups $P_1D_1D_2D_4$, $P_2D_1D_3D_4$ and $P_3D_2D_3D_4$. The parity for the first group is even parity and hence $X = 0$. The parity for the second group is not having even parity and hence $Y = 1$. The parity for the third group is even parity and hence $Z = 0$. $ZYX = 010$, whose binary equivalent is 2. Thus, the error has occurred in the second bit from MSB. Therefore, the corrected Hamming code is 1101001 and is the transmitted code.

MULTIPLE CHOICE QUESTIONS

1. The decimal equivalent of hex number 1B52 is
   (a) 6994 (b) 9649
   (c) 9964 (d) 4699
2. The base of the number 537.15_{10} is
   (a) 2 (b) 5
   (c) 10 (d) 15
3. What is the binary equivalent of the decimal number 364?
   (a) 101110000 (b) 101111110
   (c) 111010000 (d) 111100000
4. The hexadecimal decimal equivalent of $(654)_8$ is
   (a) BAC (b) 12A
   (c) 1AC (d) B1C

5. The decimal equivalent of $(159)_{16}$
   (a) 159 (b) 101
   (c) 951 (d) 345
6. The addition of 2H and CH is
   (a) 2CH (b) (2 + C)H
   (c) EH (d) None of the above
7. A megabyte represents
   (a) 1 million bytes (b) 1000 kilobytes
   (c) 100 bytes (d) 1100 bytes
8. The information stored and transmitted in a computer is in the form of
   (a) Binary (b) ASCII
   (c) Decimal (d) Alphanumeric
9. A parity bit is used to
   (a) Indicate zeros in the bit (b) Detect errors
   (c) First bit in a byte (d) Last bit in a byte
10. The binary representation of hexadecimal 'C3' is
    (a) 1111 (b) 110011
    (c) 110001 (d) 11000011
11. Gray code is NOT used in
    (a) Digital circuits (b) Encoding
    (c) Arithmetic circuits (d) K-maps
12. The punch card that used to store and encode alphanumeric data were coded in
    (a) Alphanumeric code (b) Hollerith code
    (c) Binary code (d) Excess 3 code
13. A 4-bit number is given as 1101. Its 1's complement is
    (a) 1001 (b) 11001
    (c) 0010 (d) 1101
14. In the binary number 1110.1010, the fractional part has the value
    (a) 0.50 (b) 0.25
    (c) 0.125 (d) 0.625
15. A hexadecimal number 'A2' has the decimal value
    (a) 16 (b) 32
    (c) 162 (d) 256
16. The excess -3 code equivalent for 5 is
    (a) 0011 (b) 0101
    (c) 0111 (d) 1000

17. In Gray code, for every bit added to the decimal number _____ bit changes.
    (a) 2 (b) One
    (c) No (d) All
18. The ASCII is expanded as
    (a) American Standard Code for Information Interchange
    (b) American Stock Code for Information Interchange
    (c) All Standard Code for Information Interchange
    (d) All Standard Code for Information Interlink
19. The parity for 0011 is
    (a) 0 (b) 1
    (c) Odd (d) Even
20. The threefold repetition code for 1011 is
    (a) 000100010001 (b) 100010001000
    (c) 101010101010 (d) 101110111011
21. In Hamming Code, the parity bits are always placed at positions having powers of
    (a) 1 (b) 2
    (c) 8 (d) 10

Answers

1. (a) 2. (c) 3. (b) 4. (c) 5. (d) 6. (c) 7. (b) 8. (a)
9. (b) 10. (d) 11. (c) 12. (b) 13. (c) 14. (d) 15. (c) 16. (d)
17. (b) 18. (a) 19. (c) 20. (d) 21. (b)

DESCRIPTIVE QUESTIONS

1. Convert the following decimal numbers to its binary equivalent.
    (a) 45 (b) 28
    (c) 12.4 (d) 35
    (e) 1.63
2. Convert the given decimal numbers to its octal and hexadecimal equivalent.
    (a) 637 (b) 189
    (c) 14.23 (d) 45
    (e) 2013
3. Convert the following binary numbers to its decimal equivalent.
    (a) 1010101 (b) 101.11
    (c) 10.11 (d) 1100011
    (e) 11011

4. Convert the given binary numbers to its octal and hexadecimal equivalent.
   (a) 11011011.101 (b) 1001
   (c) 01101110.10 (d) 1010.1111
   (e) 11010100
5. Convert the hexadecimal numbers to binary, decimal and octal equivalents.
   (a) 77D (b) 9E
   (c) 8CC (d) CB86
   (e) FF12
6. Perform the binary addition of the following numbers.
   (a) 1110 + 0100 (b) 1110.01 + 10.1
7. Perform binary subtraction for the following using 1's complement method.
   (a) 0.1100 – 0.0011 (b) 0.011110 – 0.100011
8. Perform binary subtraction for the following using 2's complement method.
   (a) 11110 – 10100 (b) 11001 – 00011
9. Add the hexadecimal numbers.
   (a) A5H + B6H (b) 7H + 4H
   (c) 699H + 301H
10. Perform the binary multiplication for the given numbers.
    (a) 101010 × 110 (b) 101 × 110
11. Perform the binary division for the given numbers.
    (a) 1100 by 110 (b) 10111 by 101
12. Write the excess 3 code, gray code for
    (a) 5 (b) 10
    (c) 15
13. Write the three fold repetition code for
    (a) 1101 (b) 1011
    (c) 0101
14. Write the parity and then write the parity code for
    (a) 0100100 (b) 1110001
    (c) 0111110 (d) 1111110
    (e) 1110111
15. Write the even parity hamming code for
    (a) 1110 (b) 0101
16. A 7-bit hamming code is received as 1011011. Find the error position and correct the code.
17. Write short notes on
    (a) Cyclic redundancy check code (b) Hamming code

LOGIC GATES

INTRODUCTION

Boolean algebra is a mathematical system operating on binary digits or bits specifically 0's and 1's. They perform several mathematical operations. Digital circuits that have one or more inputs, but only one output that can perform logical operations are called logic gates. The most common gates are AND, OR, and NOT, and they are written in capital letters to help differentiate the same words used in English communication. A combination of these logic gates helps to perform various logical and arithmetic functions.

Gates have binary digits as both input and output. The corresponding output is written in a tabular format for convenience. Such a table is called the truth table. The bits in the truth table are written according to the binary order. The logic gates use only two voltages, high and low. A low voltage corresponds to '0' while '1' correlates to high voltage.

POSITIVE AND NEGATIVE LOGIC

The logic gate adopts two types of systems expressly the positive logic and the negative logic systems. Consider an analog signal as shown in Figure 3.1.

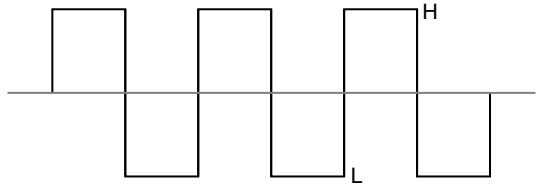


Fig.3.1 Analog signal

The higher level of the signal is designated as H and the lower level of the signal as L. H always represents 1 and L represents 0.

Positive Logic System

In this type of system, the positive voltage is taken as high level or 1, while the negative voltage is taken as low level or 0. The positive logic system is represented in Figure 3.2.

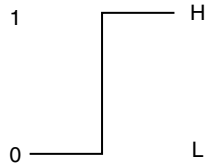


Fig.3.2 Representation of positive logic

Negative Logic System

In this type of logic system, the positive voltage is considered as low level or 0, while the negative voltage is taken as high level or 1. The negative logic system is represented in Figure 3.3.

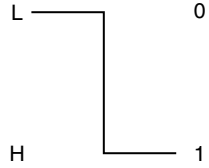


Fig.3.3 Representation of negative logic

The positive and negative do not define the signal values as there exist cases when both input and output have logic 1 or logic 0. These are just comparison between the amplitudes of the signals.

AND GATE

The AND gate is a basic logic gate that executes a multiplication operation on the given inputs. The various possible combinations of the input and output to AND gate is shown in Table 3.1. The output in case of AND gate has high output only when both the inputs are high, and in all other cases it is low which is evident from Truth Table 3.1.

Table 3.1 Truth table for a 2-input AND gate

A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

The Boolean expression for the AND operation performed on input A and B is given as:

Y = A.B or Y = AB, where (.) represents logic AND multiplication.

When

A = 0, B = 0 then Y = AB = 0.0 = 0

A = 0, B = 1 then Y = AB = 0.1 = 0

A = 1, B = 0 then Y = AB = 1.0 = 0

A = 1, B = 1 then Y = AB = 1.1 = 1

The standard symbol for an AND gate is shown in Figure 3.4. In the figure, A and B represent the inputs and Y is the output of the gate. AND gate can have any number of inputs.

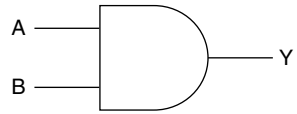


Fig.3.4 Symbol for AND gate

A 3-input AND gate has three input terminals, namely, A, B, and C. The inputs take combinations of binary digits 0 and 1. The output is verified using the truth table. A three input AND and its corresponding truth table is shown in Figure 3.5. Both input and output have same sign.

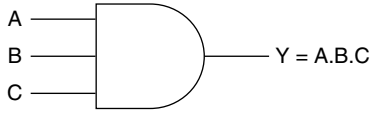


Fig.3.5 Three input AND gate—symbol and truth table

A	B	C	Y = A.B.C
0	0	0	0
0	1	0	0
0	0	1	0
0	1	1	0
1	0	0	0
1	1	0	0
1	0	1	0
1	1	1	1

The truth table begins with binary numbers 000 and finally concludes at to 111. When all the inputs are high, the output is high, and in all other cases the output is low. AND gate is enabled only when all the inputs are high. AND gate can also be explained using electronic switch circuit. The circuit utilizes two switches, S_1 and S_2.

The circuit diagram employing the switches is shown in Figure 3.6. When both the switches are closed and the current flows in the circuit, thus enabling the bulb to glow. When one switch is open, the current is interrupted and the bulb is in the off state.

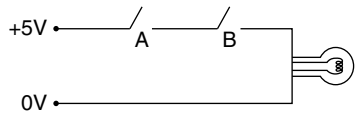


Fig.3.6 Electronic circuit to demonstrate AND operation

Thus, it is evident from Figure 3.6 that the bulb glows only when both the switches A and B are closed. When the circuit is said to be closed, here it refers to the conducting state.

OR GATE

The OR gate is a logical operator that sums up the given inputs. When two inputs A and B are fed to the OR gate, it adds both the input and displays the sum at the output. The output in case of OR gate has high output even if one input is high, and output is low when both the inputs are low. This is evident from the below Truth Table 3.2.

Table 3.2 Truth table for a 2-input OR gate

A	B	Y = A + B
0	0	0
0	1	1
1	0	1
1	1	1

The Boolean expression for the OR operation performed on input A and B is given as:

$$Y = A + B, \text{ where } (+) \text{ represents logic OR addition.}$$

When,

$$A = 0, B = 0 \text{ then } Y = AB = 0 + 0 = 0$$
$$A = 0, B = 1 \text{ then } Y = AB = 0 + 1 = 1$$
$$A = 1, B = 0 \text{ then } Y = AB = 1 + 0 = 1$$
$$A = 1, B = 1 \text{ then } Y = AB = 1 + 1 = 1$$

The standard symbol for an OR gate is shown in Figure 3.7. From the figure, A and B represent the inputs and Y is the output of the gate. AND gate can have any number of inputs.

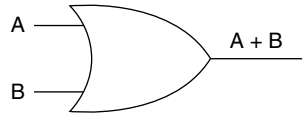


Fig.3.7 OR gate Symbol

OR gate can also the explained using electronic circuit employed with switches S_1 and S_2. The circuit is shown in Figure 3.8.

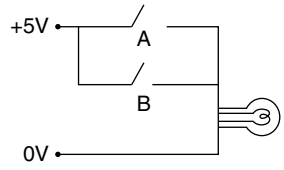


Fig.3.8 Electronic circuit to demonstrate OR gate operation

When switch A and B are open then the circuit does not conduct and current does not flow. When either A or B is open and the other is closed still the current flows across the circuit and the bulb glows.

When both A and B are closed then also the bulb glows.

A 3-input OR gate has three input terminals, namely, A, B, and C. The inputs take combinations of binary digits 0 and 1. The output is verified using the truth table. The symbol and truth table of a three input OR gate is shown in Figure 3.9.

Both the input and the output have the same sign.

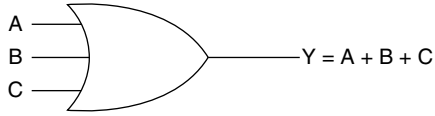


Fig.3.9 Three input OR gate symbol and truth table

A	B	C	Y = A + B + C
0	0	0	0
0	1	0	1
0	0	1	1
0	1	1	1
1	0	0	1
1	1	0	1
1	0	1	1
1	1	1	1

The truth table begins with binary numbers 000 and finally concludes at 111. It is evident from the truth table that even if one input is high the output is high and when all the inputs are low '0', the output is low. OR gate is disabled only when all the inputs are low.

NOT GATE

NOT gate is also known as the inverter. It performs inversion operation on the input, in other words, the output will be a complement of the input. It has only one input terminal as compared to OR (or) AND gates that have more than one input. The symbol of a NOT gate is denoted as shown in Figure 3.10.

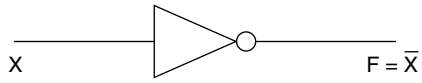


Fig.3.10 NOT gate symbol

If the input is X then the output is denoted either as X′ or X.
The truth table for a NOT gate is given as Table 3.3

Table 3.3 Truth table for NOT gate

Input (X)	Output (X′)
1	0
0	1

When the input is positive logic or high, the output is low. When the input is negative logic or low, the output is high.

The operation of the NOT gate can be explained with the help of the circuit and its corresponding truth table is shown in Figure 3.11 and Table 3.4. When the switch A is open, the current flows in the circuit and permits the bulb to glow and in the ON state. In the second case, when the switch A is closed, the current flow is obstructed and the bulb does not glow in the OFF state.

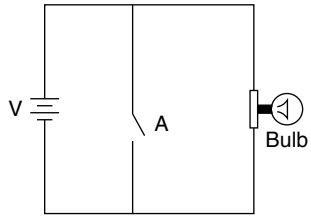


Fig.3.11

Table 3.4 NOT operation

Input (A)	Bulb
Open	ON
Closed	OFF

UNIVERSAL GATES

A universal gate is a gate which can employ any Boolean function without using the basic logic gates (AND, OR, and NOT). The NAND and NOR gates are universal gates. In addition to AND, OR, and NOT gates, other logic gates like NAND and NOR are also used in the design of digital circuits. In practice, application of universal gate is advantageous as NAND and NOR gates are economical and easier to fabricate and the basic gates used in all IC digital logic families. In fact, an AND gate is typically implemented as a NAND gate followed by an inverter not vice versa. Likewise, an OR gate is typically implemented as a NOR gate followed by an inverter not vice versa.

NAND AS A UNIVERSAL GATE

The NAND gate is a popular universal logic element as it is used in combination to perform the AND, OR, and inverter operations. The term NAND is a short form of NOT + AND. NAND gate is combination of AND gate followed by NOT gate. In other words, it performs inversion on the output of AND gate. The standard logic symbol for a 2-input NAND gate and its equivalency to an AND gate followed by an inverter are shown in Figure 3.12.

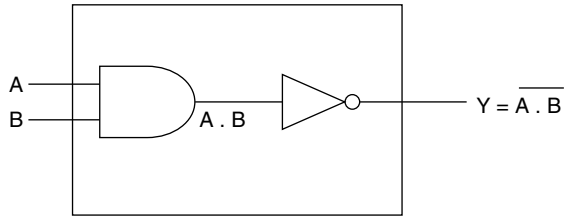


Fig.3.12 Standard NAND gate logic symbols

OPERATION OF A NAND GATE

When all the inputs are HIGH, the NAND gate produces a LOW output. The output will be HIGH even when any of the inputs is LOW. Consider a 2-input NAND gate, as shown in Figure 3.13 with the inputs A and B and the output X, the operation can be stated as follows: For a 2-input NAND gate, when inputs A and B are HIGH, the output X is LOW; when either A or B is LOW or when both A and B are LOW, then the output X is HIGH. This shows that the operation of NAND is reverse of AND in terms of the output level. Figure 3.13 illustrates the operation of a 2-input NAND gate for all four input combinations. The Boolean expression for a 2-input NAND gate with inputs A and B are given as:

$$X = \overline{A.B}.$$

The above expression holds good when the inputs A and B are first ANDed and complemented as represented by the bar indicated over the AND operation. The truth table for a 2-input NAND gate is shown in Table 3.5.

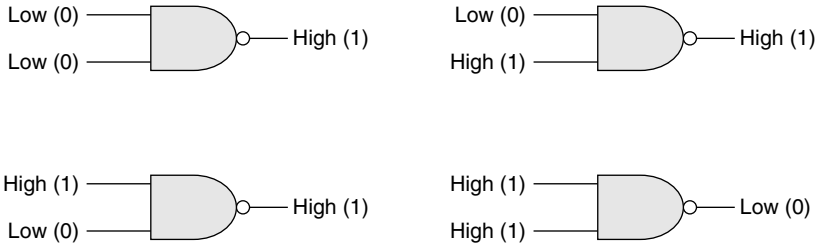


Fig.3.13 Operation of a 2-input NAND gate

Table 3.5 Truth table of a 2-input NAND gate

Inputs		Output
A	**B**	**$X = \overline{A.B}$**
0	0	$\overline{0.0} = \overline{0} = 1$
0	1	$\overline{0.1} = \overline{0} = 1$
1	0	$\overline{1.0} = \overline{0} = 1$
1	1	$\overline{1.1} = \overline{1} = 0$

NAND gate can have any number of inputs. Consider a three inputs NAND gate. The symbol for a 3-input NAND gate is as shown in Figure 3.14 and truth table of a 3-input NAND gate is shown in Table 3.6.

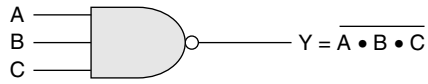


Fig.3.14 3-input NAND gate

Table 3.6 Truth table for 3-input NAND gate

A	B	C	(A.B.C)	$Y = \overline{A.B.C}$
0	0	0	0	1
0	1	0	0	1
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	1	0	0	1
1	0	1	0	1
1	1	1	1	0

NAND gate for three inputs gives high value only when the product of all the three inputs are low and high for all other combinations of the inputs.

Apart from the basic logic gates like OR, AND, and NOT, these gates are a fusion of OR (or) AND with NOT and are useful in facilitating desired designs of digital circuits. The NAND gate can be implemented to prove any Boolean function. Also NAND gate can be used to replace OR, AND, and NOT gates in a circuit thus minimizing the components in any given circuit. The implementation of NAND gates will be dealt elaborately in Chapter 6 of this book.

Negative OR-Equivalent Operation of a NAND Gate

The truth table of a NAND gate exhibits that a HIGH output is produced by one or more LOW inputs. Therefore, it is concluded from the truth table that a NAND gate can be used for an OR operation that requires one or more LOW inputs to produce

a HIGH output. This characteristic of NAND operation is referred to as negative-OR. The term *negative* is used since the inputs are LOW. Consider a 2-input NAND gate performing a negative-OR operation, when either input A or input B is LOW or when both A and B are LOW, the output X is HIGH. Thus, a NAND gate working in the negative-OR operation can be used to detect one or more LOWs on its inputs. The standard symbols representing the two equivalent operation of a NAND gate are shown in Figure 3.15.

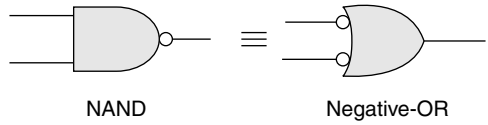


Fig.3.15 Standard symbols representing the two equivalent operations of a NAND gate

NOR GATE

A NOR gate is the next widely used universal gate. It finds its importance in many circuits and can replace the AND, OR, and NOT logic operators. The NOR gate is an outcome of the combination of OR gate followed by a NOT gate (OR + NOT = NOR). This gate performs OR operation on the input and then inverts it to give the final output. The standard logic symbol for a 2-input NOR gate and its equivalent OR gate followed by an inverter is shown in Figure 3.16. Boolean functions can be proved using a NOR gate without using the basic gates. NOR gate is similar to the NAND gate. It can perform operations of all other gates and hence rightly termed as a universal gate.

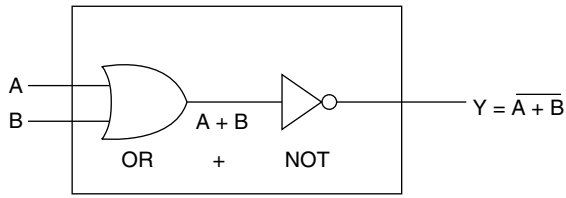


Fig.3.16 NOR equivalent logic symbol

NOR gate can also be expressed as the complement of OR gate. Consider a 2-input NOR gate, the circuit is shown in Figure 3.17 and Truth Table 3.7.

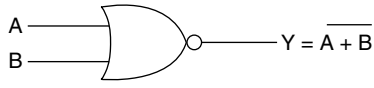


Fig.3.17 2-input NOR gate

Table 3.7 Truth table of 2-input NOR gate

A	B	(A + B)	$Y = \overline{A + B}$
0	0	0	1
1	0	1	0
0	1	1	0
1	1	1	0

OPERATION OF NOR GATE

Consider a 2-input NOR gate, when one of the inputs of a NOR gate is HIGH, then the output produced is LOW. The output is HIGH only when all of its inputs are LOW. When A is HIGH and B is HIGH the output is LOW. When A is LOW and B is HIGH or when A is HIGH and B is LOW then also the output still remains LOW. The output is HIGH only when both A and B are LOW. This proves that the operation of NOR gate is opposite to that of the OR gate. The bubble denotes that the LOW output is active or asserted output level. The circuit for a 3-input NOR gate is shown in Figure 3.18. Truth table for a 3-input NOR gate is shown in Table 3.8.

Logic Expressions for a NOR Gate

The Boolean expression for the output of a 2-input NOR gate can be written as

$$X = \overline{A + B}$$

From the equation, it is understood that the 2-input variables are first ORed and then an inversion is performed which is indicated by the bar over the OR expression.

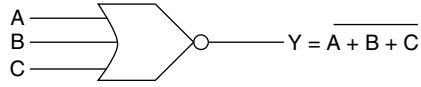


Fig.3.18

Table 3.8 Truth table for 3-input NOR gate

A	B	C	(A + B + C)	$Y = \overline{A + B + C}$
0	0	0	0	1
0	1	0	1	0
0	0	1	1	0
0	1	1	1	0
1	0	0	1	0
1	1	0	1	0
1	0	1	1	0
1	1	1	1	0

The operation of a 3-input NOR gate is similar to 2-input NOR gate and it is evident from its truth table.

Negative-AND Equivalent Operation of the NOR Gate

Consider the truth table of a NOR gate. It is evident from the truth table that the NOR output is HIGH only when both the inputs are LOW, and are LOW for all other input combinations. Thus, a NOR gate can be used for an AND operation that requires all LOW inputs to produce a HIGH output. This aspect of NOR operation is called negative-AND. The term *negative* is used to denote that the inputs are active when LOW. The NOR gate while performing the negative-AND operation is used to detect if all LOW is on its inputs. The standard symbol for negative NOR gate is shown in Figure 3.19.

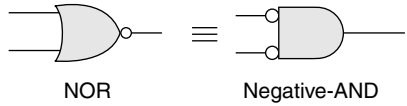


Fig.3.19 Standard symbols representing the two equivalent operations of a NOR gate

THE EXCLUSIVE-OR AND EXCLUSIVE-NOR GATES

The exclusive-OR and exclusive-NOR gates are formed by a combination of other gates that are already discussed. They find their importance in many digital circuits and hence are treated as yet another gate with their own unique symbols.

XOR Gate

An exclusive-OR gate also abbreviated as XOR has two inputs and one output. Standard symbol for an exclusive-OR gate is shown in Figure 3.20. The XOR gate has two inputs, A and B. The output is denoted as Y.

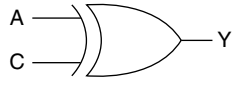


Fig.3.20 XOR gate logic symbol

The equivalent symbol for a XOR gate is shown in Figure 3.21.

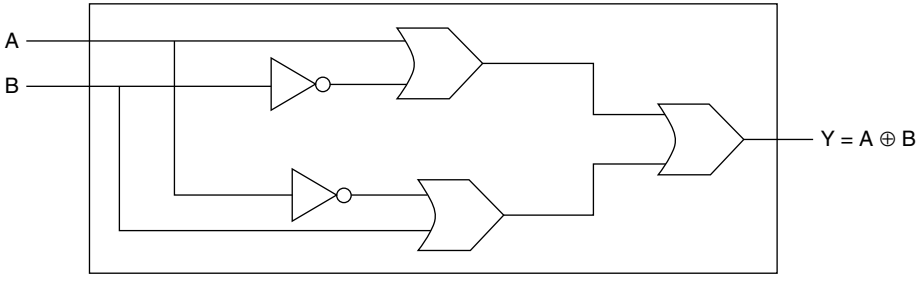


Fig.3.21 XOR gate equivalent symbol

OPERATION OF XOR GATE

When either of the inputs is LOW then the output is HIGH. However when both the inputs A and B are HIGH or LOW the output Y is LOW. The four possible input combinations and the resulting outputs for a XOR gate are illustrated in Figure 3.22.

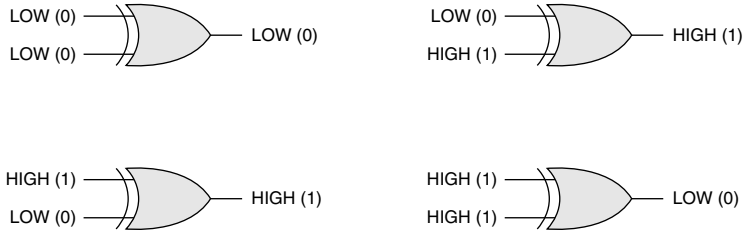


Fig.3.22 Operation of XOR gate

The truth table of XOR gate is given in Table 3.9.

Table 3.9 Truth table for 2-input XOR gate

A	B	A′	B′	AB′	A′B	Y = AB′ + A′B
0	0	1	1	0	0	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	0	0	0

EX-NOR GATE (EXNOR/XNOR)

XNOR gate is a combination of a XOR gate followed by a NOT gate. Like the XOR, the XNOR also has two inputs. The Boolean expression for an XNOR gate is

$$Y = \overline{A \oplus B}$$

The standard symbol for XNOR gate is shown in Figure 3.23

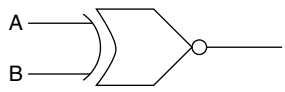


Fig.3.23 Standard XNOR symbol

The equivalent circuit for XNOR gate is shown in Figure 3.24

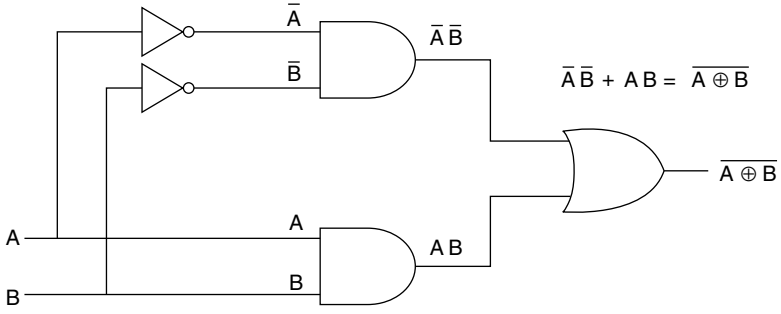


Fig.3.24 Equivalent circuit for XNOR

Operation of XNOR Gate

Consider a XNOR gate with inputs A and B. The output is denoted as Y. When one of the two inputs is either HIGH or the other is LOW, then the output of the exclusive-NOR gate is LOW. For an exclusive-NOR gate, output Y is LOW when input A is LOW and input B is HIGH, or when A is HIGH and B is LOW; Y is HIGH when A and B are both HIGH or both LOW. The four possible input combinations and the resulting outputs for an XNOR gate are shown in Figure 3.25. The truth table for an XNOR gate is shown in Table 3.10. It is evident from the table that the output is HIGH when both the inputs have the same level.

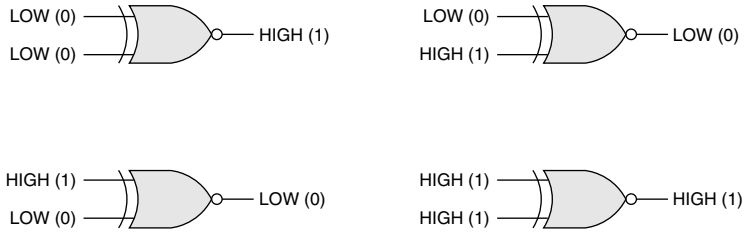


Fig.3.25 Operating of XNOR gate

Table 3.10 Truth table for 2-input XNOR

A	B	A′	B′	AB′	A′B	$Y = AB' + A'B$	$Y = \overline{A \oplus B}$
0	0	1	1	0	0	0	1
0	1	1	0	0	1	1	0
1	0	0	1	1	0	1	0
1	1	0	0	0	0	0	1

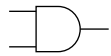
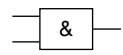
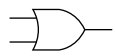
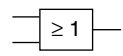
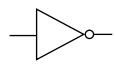
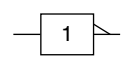
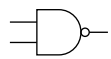
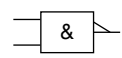
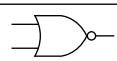
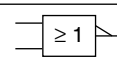
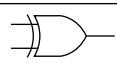
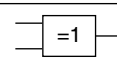
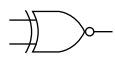
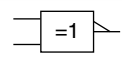
Thus, the XNOR has high output, only when both the inputs are either high or low.

IEEE SYMBOLS

The American National Standards Institute (ANSI) and the Institute of Electrical and Electronic Engineers (IEEE) jointly developed a standard set of logic symbols. They are the IEEE Standard Graphic Symbols for Logic Functions. It is compatible with the International Electrotechnical Commission (IEC), and must be used in all logic diagrams. The IEEE standard provides two different types of symbols for logic gates. One type, called distinctive-shape symbols, which are traditionally used and the other type, called rectangular-shape symbols, uses the same shape for all the gates, along with an internal label to identify the type of gate. The logic gate symbols referred by IEEE and traditional symbols are shown in Table 3.11.

The objective of IEEE was to provide a uniform method of describing the complex logic functions of digital circuits with schematic symbols. In electronics, a NOT gate is more commonly called an inverter. The circle on the symbol is called a *bubble*, and is used in logic diagrams to indicate a logical inversion between the output and the input. On a circuit diagram, it must be accompanied by a statement asserting that the positive logic convention or negative logic convention is being used (high voltage level = 1 or high voltage level = 0, respectively). The *wedge* is used in circuit diagrams to directly indicate an active-low input or output (high voltage level = 0) without performing a convention throughout the circuit diagram. This is called direct polarity indication. Both the bubble and the wedge can be used on distinctive-shape and rectangular-shape symbols on circuit diagrams, depending on the logic convention used.

Table 3.11 IEEE symbol

Logic function	Traditional symbols	IEEE symbols	Boolean expression
AND			$A \cdot B$
OR			$A + B$
NOT			$\overline{A}$
NAND			$\overline{A \cdot B}$
NOR			$\overline{A + B}$
XOR			$A \oplus B$
XNOR			$\overline{A \oplus B}$

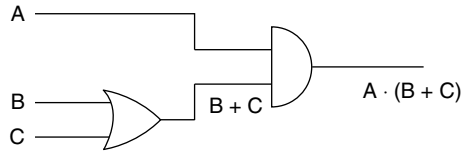
ADDITIONAL EXAMPLES

Example 3.1 Represent the Boolean identity using basic gates.

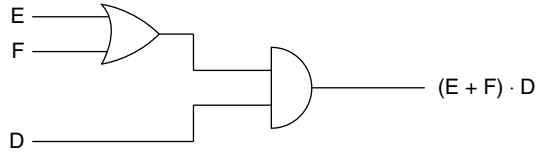
$$Y = A \cdot (B + C) + (E + F) \cdot D$$

Solution: The above Boolean expression can be implemented using basic logic gates as follows:

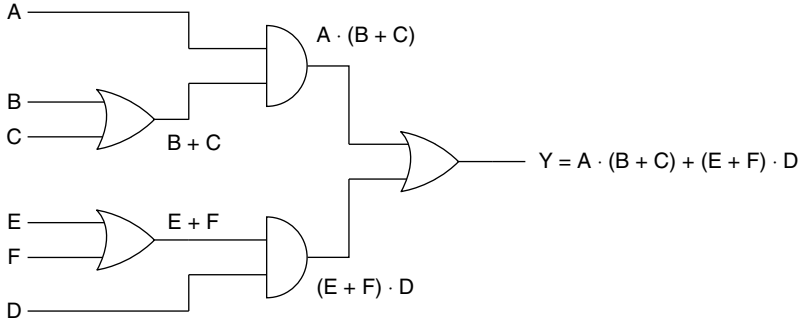
1st term: $A \cdot (B + C)$



2nd term: $(E + F) \cdot D$



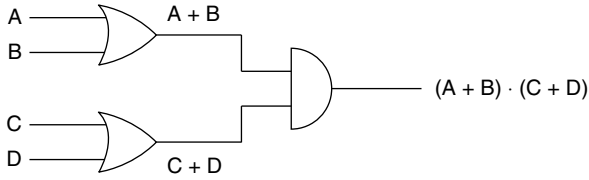
Combining both, we get the desired output.



Example 3.2 Implement the Boolean expression using basic gates.

$$Y = (A + B) \cdot (C + D)$$

Solution:



Example 3.3 Implement the Boolean expressions using logic gates.

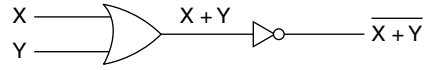
(i) $\overline{X+Y} = \overline{X} \cdot \overline{Y}$

(ii) $\overline{X \cdot Y} = \overline{X} + \overline{Y}$

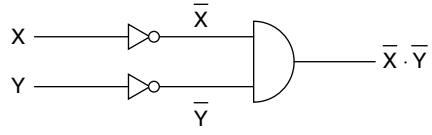
Solution:

(i) $\overline{X+Y} = \overline{X} \cdot \overline{Y}$

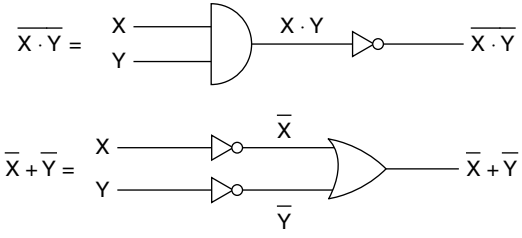
L.H.S. Logic representation for $\overline{X+Y}$



R.H.S.



(ii) $\overline{X \cdot Y} = \overline{X} + \overline{Y}$

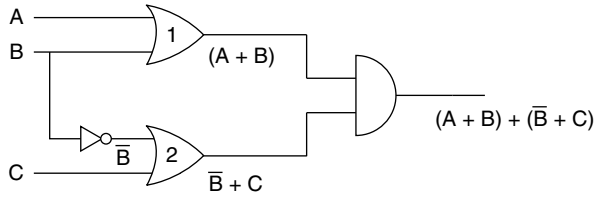


Hence, the above expressions are represented using logic gates.

Example 3.4 Depict the Boolean expression $Y = (A+B)(\overline{B}+C) = A\overline{B} + BC$ using gates.

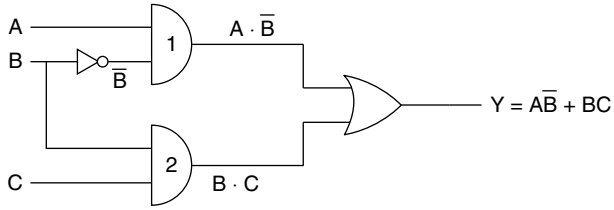
$$Y = (A+B)(\overline{B}+C) = A\overline{B} + BC$$

Consider the L.H.S., the logic diagram is implemented as:



The output of OR gate 1 is (A + B). This is fed as one of the input to the AND gate. The other input to the AND gate is given from OR gate 2. $\overline{B}$ is obtained by passing B through an inverter (or) NOT gate.

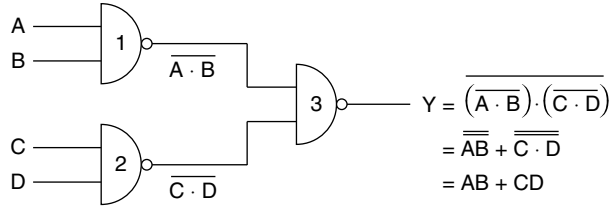
R.H.S.



The inputs to the AND gate 1 are A and $\overline{B}$. $\overline{B}$ is obtained by passing B to NOT gate. The output of AND gate 1 is A.$\overline{B}$. The inputs to the second AND gate are B and C. Thus, the output is B.C. The final output is obtained at the OR gate.

Example 3.5 Implement the Boolean expression Y = AB + CD using only NAND gate.

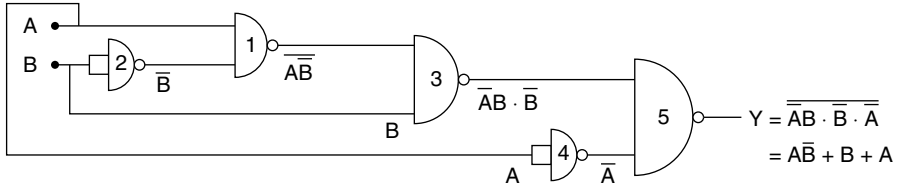
Solution: The logic expression for the Boolean expression is depicted as below.



NAND gate 1 and 2 are used to get outputs $\overline{A \cdot B}$ and $\overline{C \cdot D}$. These two outputs are fed to the NAND gate 3 which gives the final output.

Example 3.6 Design logic diagram for the Boolean expression Y = A$\overline{B}$ + B + A using NAND gates only.

Solution: The logic diagram is implemented as



The above Boolean can be implemented using logic gates in three steps.

Step 1: Consider the first term $A\overline{B}$, to obtain this the inputs of the NAND gate 1 is A and $\overline{B}, \overline{B}$ which is obtained by using NAND as an inverter.

Step 2: The output of NAND gate 1 and B form the input for gate "3". The output of NAND gate 3 is $\overline{A}B \cdot \overline{B}$.

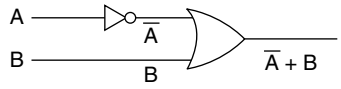
Step 3: The output of NAND gate "3" i.e., $\overline{A}B \cdot \overline{B}$ and output of gate "4" forms the input for gate "5". Gate "5" gives the final desired output.

$$Y = A\overline{B} + B + A$$

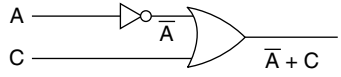
Example 3.7 Implement the Boolean expression $Y = (\overline{A} + B) \cdot (\overline{A} + C)$ using basic gates.

Solution: $$Y = (\overline{A} + B) \cdot (\overline{A} + C)$$

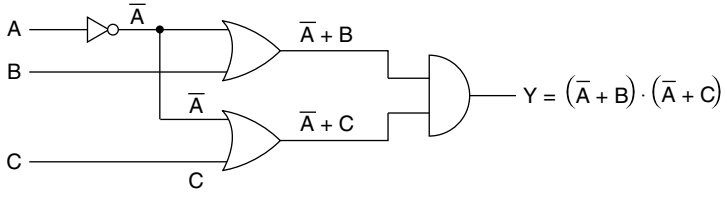
Consider the first term $\overline{A} + B$. It can be expressed logically as



The second term is represented as



Combining both, we get

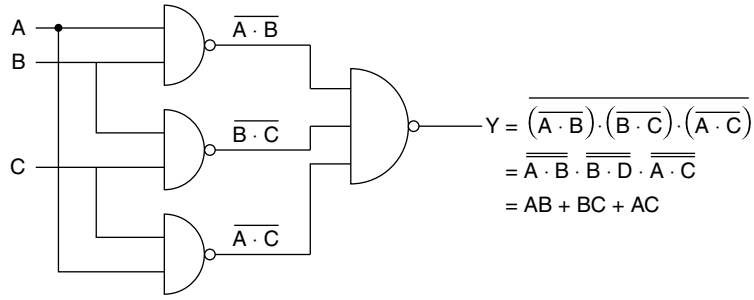


Hence, the logic diagram is implemented.

Example 3.8 Implement the Boolean expression using NAND gates.

$$Y = AB + BC + AC$$

Solution:

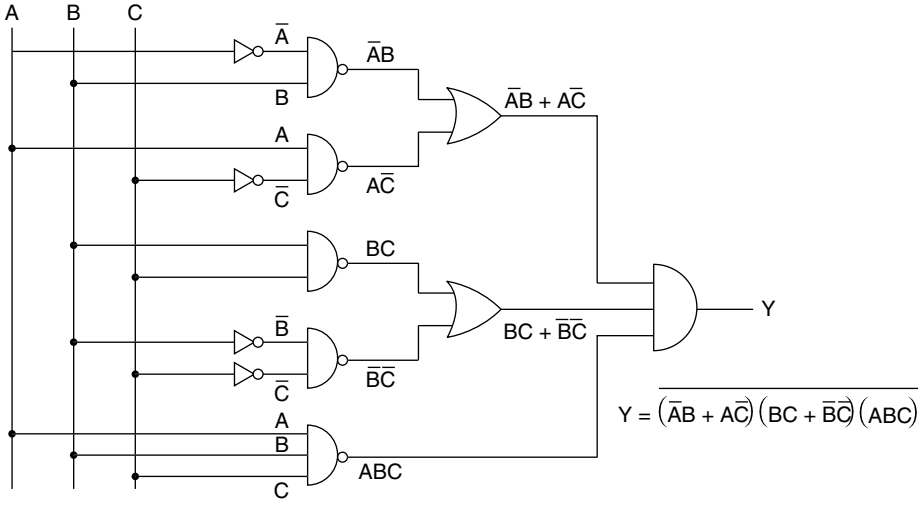

A
B
C
$\overline{A \cdot B}$
$\overline{B \cdot C}$
$\overline{A \cdot C}$
$Y = \overline{(\overline{A \cdot B}) \cdot (\overline{B \cdot C}) \cdot (\overline{A \cdot C})}$
$= \overline{\overline{A \cdot B} \cdot \overline{B \cdot D} \cdot \overline{A \cdot C}}$
$= AB + BC + AC$

Thus, the above expression is implemented using only NAND gate. This ascertains that NAND is a universal gate.

Example 3.9 Express the Boolean expression using basic gates.

$$Y = \left(\overline{A}B + A\overline{C}\right)\left(BC + \overline{B}\,\overline{C}\right)\left(ABC\right)$$

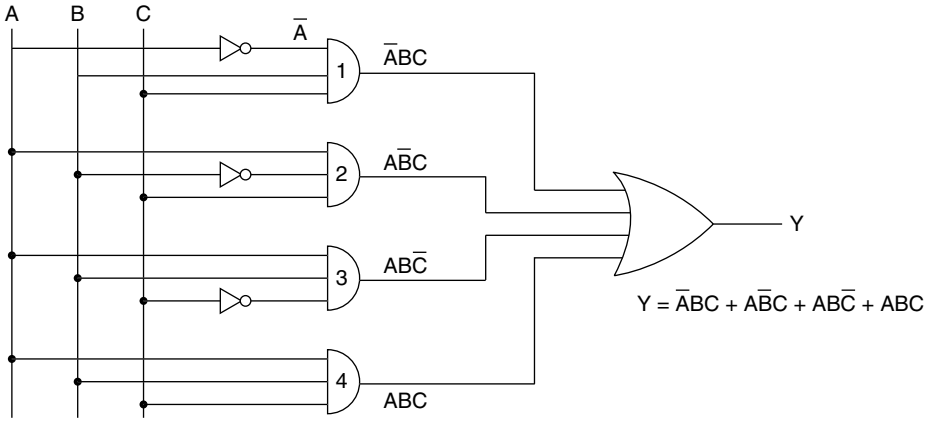
Solution:



Example 3.10 Implement the Boolean expression using basic gates

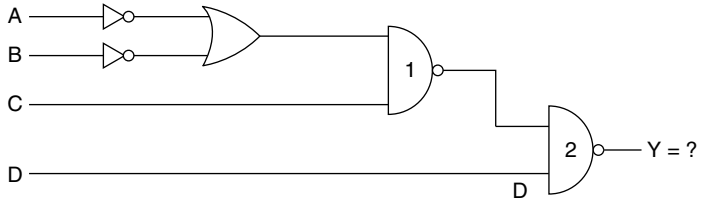
$$Y = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

Solution:



The Boolean expression can be implemented using 4 AND gates and 1 OR gate. The output of all 4 AND gates are fed as input to the OR gate to get the desired Boolean expression.

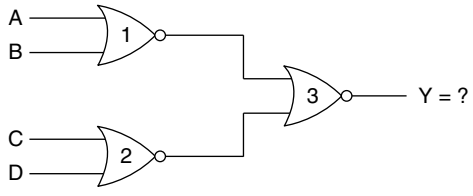
Example 3.11 Deduce the Boolean expression from the logic diagram.



Solution: The inputs to the OR gate is $\overline{A}$ and $\overline{B}$. Thus, $\overline{A+B}$ from the OR gate is fed as the input to NAND gate 1. The output of NAND gate "1" is $\left(\overline{A+B}\right)\cdot C$. This is fed to the NAND gate 2. The output of NAND gate 2 gives the final Boolean expression.

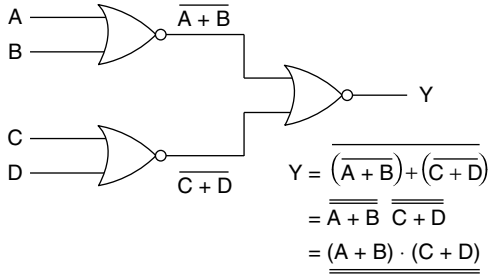
$$Y = \left(\overline{A+B}\right)\cdot C + \overline{D}$$

Example 3.12 Derive the expression from the given logic diagram.



Solution: The logic diagram has 3 NOR gates. The input of the gate 1 is A and B, its output is $\overline{A+B}$. The input to the 2nd gate is C and D, its output will be $\overline{C+D}$ as the NOR gates performs addition and later inverts the result. The input to gate 3 is from the output of gates 1 and 2. Therefore, the final output is obtained at gate 3 as

$$\begin{aligned} Y &= \overline{\overline{A+B}+\overline{C+D}} \\ &= \overline{\overline{(A+B)}+\overline{(C+D)}} \\ &= (A+B)\cdot(C+D) \end{aligned}$$



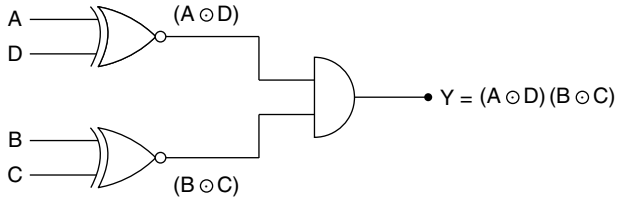
Example 3.13 Implement the Boolean expression using only 3 gates.

$$Y = ABCD + A\bar{B}CD + \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D}$$

Solution: To implement the given Boolean expression, we need to reduce the expression.

$$\begin{aligned} Y &= ABCD + A\bar{B}\bar{C}D + \bar{A}BC\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D} \\ &= AD\left(BC + \bar{B}\bar{C}\right) + \bar{A}\bar{D}\left(BC + \bar{B}\bar{C}\right) \\ &= \left(AD + \bar{A}\bar{D}\right)\left(BC + \bar{B}\bar{C}\right) \\ &= \left(A \odot D\right)\left(B \odot C\right) \end{aligned}$$

This can be represented using two XNOR gates and one AND gate.



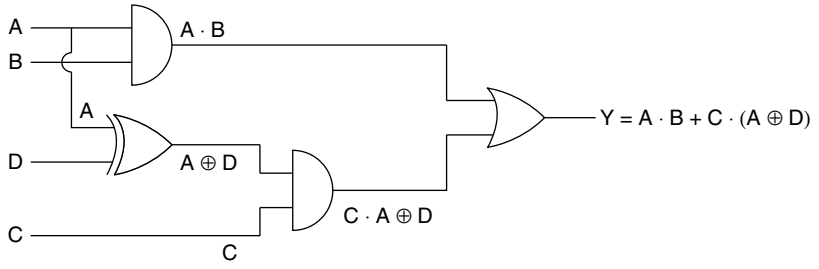
Example 3.14 Implement the Boolean expression using logic gates.

$$Y = AB + AC\overline{D} + \overline{A}CD$$

Solution:

$$\begin{aligned} Y &= AB + AC\overline{D} + \overline{A}CD \\ &= AB + C\left(A\overline{D} + \overline{A}D\right) \\ Y &= A \cdot B + C \cdot \left(A \oplus D\right) \end{aligned}$$

The above expression can be designed using 2 AND gates, one "OR" gate and one "XOR" gate.



Example 3.15 Implement the Boolean expression using minimum number of gates.

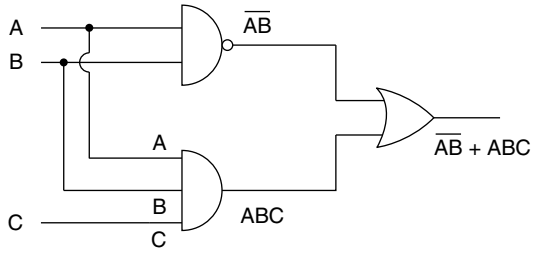
(i) $Y = \overline{AB} + ABC$

(ii) $Y = \overline{A} + \overline{B}C$

Solution:

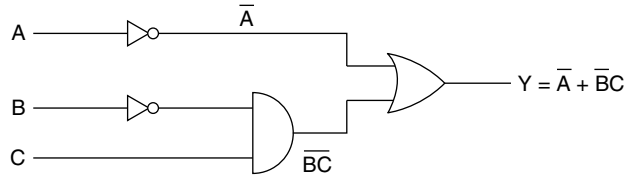
(i) $Y = \overline{AB} + ABC$

The above expression can be represented using NAND gate, OR gate. The NAND gate has an input A and B that gives output $\overline{AB}$. This is ORed to the output of AND gate to give the final output. $Y = \overline{AB} + ABC$ is shown in the logic diagram.



(ii) $Y = \overline{A} + \overline{B}C$

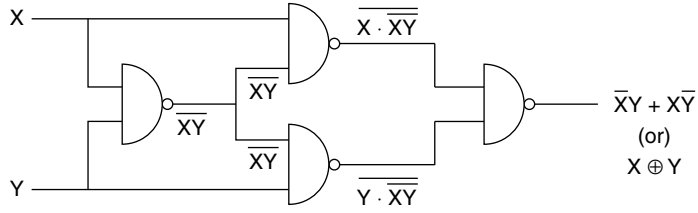
The above Boolean expression can be implemented by logic diagram as shown below:



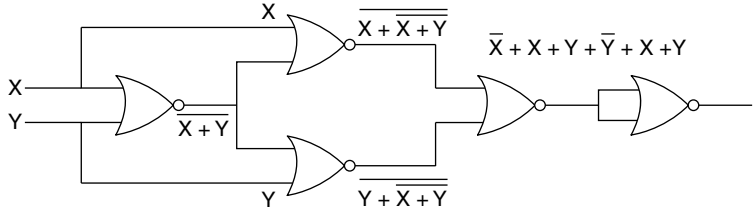
Input A after passing through the inverter gives $\overline{A}$. Input B is passed through the inverter to obtain $\overline{B}$ and is passed to AND gate whose other input is C. The output of the AND gate is $\overline{B}C$. Then $\overline{A}$ and $\overline{B}C$ are fed to OR gate which sums them to give the final output.

Example 3.16 Implement $X \oplus Y$ using (i) NAND gates only (ii) NOR gates only.

(i) NAND implementation:



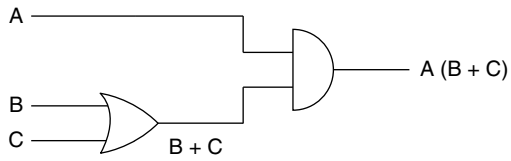
(ii) Using NOR gate:



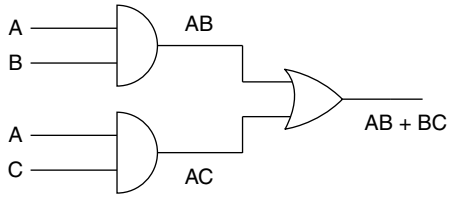
Example 3.17 Illustrate the Boolean expression $A(B + C) = AB + BC$ using basic logic gates.

Solution: $A(B + C) = AB + BC$

L.H.S.: The variable A is ANDed to (B + C). First variables B and C are ORed, to this A is ANDed. Therefore, A(B + C) is depicted as follows:

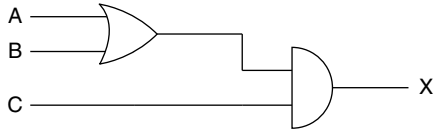


R.H.S.: The first term is obtained by ANDing variables A and B. The second term is obtained by passing variables A and C. Both these outputs are ORed.

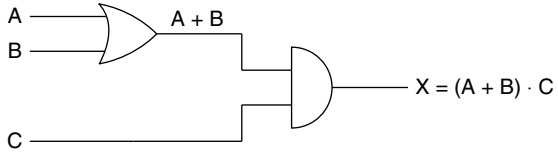


Hence, the Boolean expression is depicted.

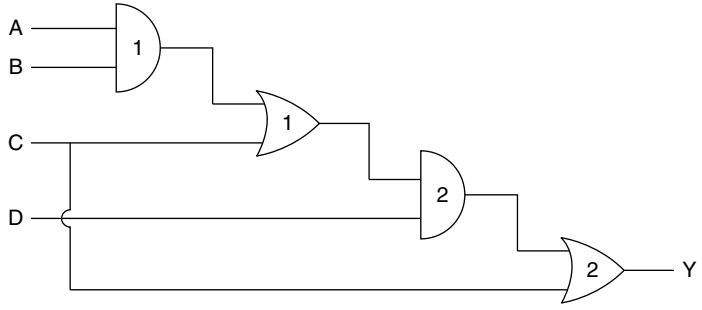
Example 3.18 Find the output of the logic diagram.



Solution: The inputs of the OR gate are A and B. Its output is given as (A + B). The output of the OR gate is fed as the input to the AND gate whose other input is C. Therefore, the final output is given as X = AB + C.

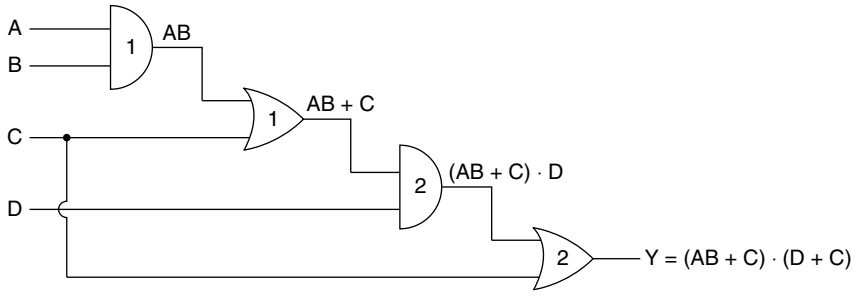


Example 3.19 Deduce the output of the logic diagram



Solution: Consider the AND gate 1. The input is A and B, its output is AB. The output from the first gate becomes the input to OR gate 1. The other input to the OR gate is C. Therefore, the output of the OR gate is given as AB + C. This output is fed to the input of the AND gate 2 whose other input is D. Therefore, the output of the AND gate 2 is $(AB + C) \cdot D$. This is given as the input to OR gate 2 whose other input is D. Therefore, the final output is given as

$$Y = (AB + C) \cdot D + C$$

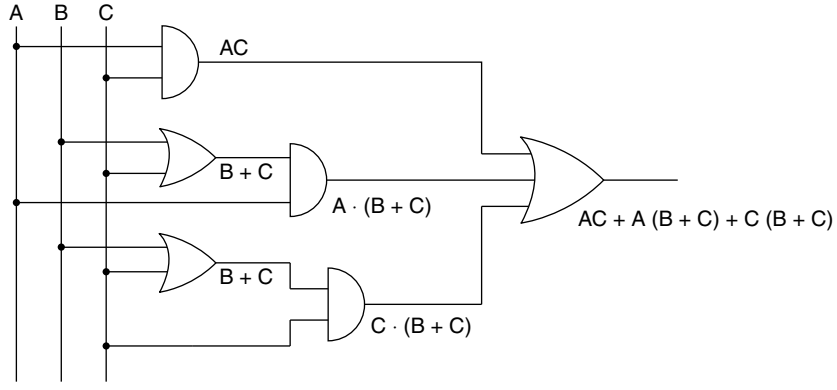


Example 3.20 Prove that the Boolean expression can be reduced and also prove that minimum logic gates are used for logical expression.

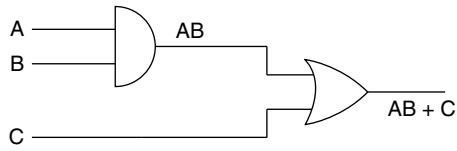
$$Y = AB + A(B + C) + C(B + C)$$

Solution: The given Boolean expression is reduced as follows:

$$\begin{aligned} Y &= AC + A(B + C) + C(B + C) \\ &= AC + AB + AC + BC + CC \quad \text{since } C \cdot C = C \\ &= AC + AC + AB + BC + C \quad \text{since } AC + AC = AC \\ &= AC + AB + BC + C \quad BC + C = C \\ &= AC + AB + C \\ &= AB + AC + C \quad \text{since } AC + C = C \\ &= AB + C \end{aligned}$$



Reduced logic expression is



MULTIPLE CHOICE QUESTIONS

1. If A, B and C are the inputs of AND gate, then the output Y is HIGH for
   (a) A = 1, B = 1, C = 0 (b) A = 0, B = 0, C = 0
   (c) A = 1, B = 1, C = 1 (d) A = 1, B = 0, C = 1
2. NAND gate is
   (a) AND followed by NOT (b) NOT followed by AND
   (c) Two AND gates interconnected (d) OR followed by AND
3. If one of the inputs of the 2-input logic gate is LOW, then which of the following gate still has an HIGH output is HIGH?
   (a) AND (b) NAND
   (c) NOR (d) OR
4. The Boolean expression for a 3-input AND gate is
   (a) Y = AB + BC + CA (b) Y = ABC
   (c) Y = A + B + C (d) Y = AB + C
5. Which of the following logical operations has '+' as its symbol?
   (a) AND (b) NAND
   (c) OR (d) NOT
6. If A, B, C are the three inputs of an OR gate, then the output is LOW when
   (a) A = 0, B = 0, C = 1 (b) A = 0, B = 1, C = 1
   (c) A = 0, B = 0, C = 0 (d) All of the above
7. The logic expression for a NOR gate is given as
   (a) $X = \overline{A} + B$ (b) $X = A + \overline{B}$
   (c) $X = A + B$ (d) $X = \overline{A + B}$
8. The NOR gate has a HIGH output when
   (a) All inputs are HIGH (b) Any one input is HIGH
   (c) Any one input is LOW (d) All inputs are LOW
9. The Boolean expression for the output of a NOR gate is similar to a
   (a) NAND gate followed by an inverter
   (b) OR gate followed by an inverter

(c) AND gate followed by an inverter
(d) NOR gate followed by an inverter

10. A NOT gate has a HIGH output when
(a) The input is LOW
(b) The input is HIGH
(c) Power supply is ON
(d) Power supply is OFF

11. An exclusive-OR gate has a HIGH output when
(a) All inputs are LOW
(b) All inputs are HIGH
(c) The inputs have opposite logic
(d) None of the above

12. Which of the following gates are interchangeable?
(a) NAND and NOR
(b) NOR and OR
(c) AND and OR
(d) NOR and bubbled OR

13. A 2-input NOR gate is equivalent to a
(a) Negative-OR gate
(b) Negative-AND gate
(c) Negative-NAND gate
(d) None of the above

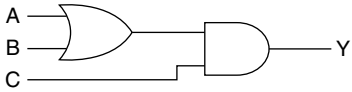
14. The following truth table is that of an

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

(a) OR
(b) AND
(c) NOR
(d) NAND

15. The Boolean expression for XNOR is given as
(a) $A + B$
(b) $A \cdot B$
(c) $A \oplus B$
(d) $\overline{A \oplus B}$

16. The output 'Y' of the given logic diagram is



(a) $Y = (A + B)\,C$
(b) $Y = AB + C$
(c) $Y = A + B + C$
(d) $Y = A + BC$

Answers

1. (c) 2. (a) 3. (b) 4. (b) 5. (c) 6. (c) 7. (d) 8. (d)
9. (b) 10. (a) 11. (c) 12. (a) 13. (b) 14. (b) 15. (d) 16. (a)

DESCRIPTIVE QUESTIONS

1. What is a XOR gate? Give its construction using only NAND gates.
2. Explain the working of a 2-input OR gate and substantiate with suitable diagrams.
3. Justify NAND as a universal building block.
4. What is a universal gate? Give examples. Realize the basic gates with universal NOR gate.
5. What is a XNOR gate? Give its truth table and equivalent circuit.
6. Define positive logic and negative logic states.
7. Discuss the importance of standard and IEEE symbols used in logic circuits.
8. Explain the working of a 2-input AND gate with suitable diagrams.
9. Represent the given Boolean expression using basic logic gates.
   (a) $Y = A + BC$ (b) $Y = AB + BC$
10. Implement the Boolean expression $Y = AB + CD + AD'$
11. Implement the Boolean expression $Y = AC + AB$ using only NAND gates.

Simplify the Boolean expressions and implement using minimum number of gates.

12. $Y = ABC + \overline{A}BC + A\overline{B}C + AB\overline{C} + \overline{A}\overline{B}\overline{C}$
13. $Y = \overline{A}B\overline{C}\overline{D} + AB\overline{C} + BC\overline{D} + \overline{A}CD + A\overline{B}D$
14. $Y = \overline{A}B + A\overline{B} + AB$
15. $Y = A\overline{B} + B\overline{C} + \overline{A}$

Reduce the following Boolean expressions

16. $AB + A(\overline{B} + \overline{C})$
17. $XX + YY$
18. $\overline{\overline{A\overline{B}} + \overline{AC} + A}$
19. $AB + \overline{AC} + A\overline{B}C$
20. $\overline{\overline{A} + B + \overline{A\overline{B}}}$
21. $A + B\overline{(C + \overline{DA})}$

BOOLEAN ALGEBRA

Digital circuits operate with two binary digits, 0 and 1. These logic circuits perform various arithmetic operations on the given inputs. Boolean, a mathematician, proposed an algebra that can be used to express these logic functions called the Boolean algebra. It comprises of a set of rules and theorems that governs the operation of the logic gates. As in case of algebra, alphabets are used to denote variables.

FUNDAMENTAL SYMBOLS—BOOLEAN ALGEBRA

There are four connecting symbols used in Boolean algebra.

1. Equal sign (=)
2. Multiplication sign (·)
3. Plus sign (+)
4. Inversion sign (‘) or ($\bar{}$)

1. **Equal sign:** This refers to the sign of equality as in mathematics. The logical value on one side is the same as the logical value on the other side.

   E.g., A = B; if A = 1 and B = 1, then the value on the left of the expression is the same as in the right side.
2. **Multiplication sign:** It refers to the AND operation.

   E.g., A · B = 0; AND operator multiplies input A with input B, here the values of A and B can either be same or different.
3. **Plus sign:** This refers to the OR operation.

   E.g., A + B = 1; OR operator adds A with B, here the values of A and B can either be same or different.
4. **Inversion sign:** This operation performs a complement of the input given to the logic gate.

   E.g., Y = A’; if A equals to 1, then Y = 0 and if A equals to 0, then Y = 1.

BOOLEAN ALGEBRAIC PROPERTIES

There are numerous Boolean properties that explain and simplify the construction of digital circuits. The following are the properties:

- **Commutative property:** This property elucidates that the output is the same even when the input is reversed. The OR and AND operators for the commutative property can be written as in Table 4.1.

Table 4.1 Commutative property and Boolean expression

Commutative property	Boolean expression	Examples
OR/Addition	$X + Y = Y + X$	$1 + 0 = 0 + 1 = 1$
AND/Multiplication	$XY = YX$	$1 \cdot 0 = 0 \cdot 1 = 0$

The truth table in Table 4.2 for 2-input commutative property is as follows:

Table 4.2 Truth table for commutative property

A	B	AB	BA	A + B	B + A
0	0	0	0	0	0
1	0	0	0	1	1
0	1	0	0	1	1
1	1	1	1	1	1

It is evident from the truth table that the commutative property holds true for both OR and AND operators.

- **Associative property:** This property suggests that the order of performing an OR operation does not change the output. The associative property of an OR and AND operator is given in Table 4.3.

Table 4.3 Associative property for OR and AND operators

Associative property	Boolean expression
OR/Addition	$X + (Y + Z) = (X + Y) + Z$
AND/Multiplication	$(X \cdot Y) \,.\, Z = X \cdot (Y \cdot Z)$

Truth table for the verification of associative rule for OR operator is shown in Table 4.4.

Table 4.4 Truth table for associative property of an OR operator

X	Y	Z	X + Y	Y + Z	X + (Y + Z)	(X + Y) + Z
0	0	0	0	0	0	0
0	1	0	1	1	1	1
0	0	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	1	0	1	1
1	1	0	1	1	1	1
1	0	1	1	1	1	1
1	1	1	1	1	1	1

- **Distributive property:** This property explains the product of the sums and the sum of the products. It is the combination of both commutative and associative properties of additive and multiplication operations. The distributive property is shown in Table 4.5.

Table 4.5 Boolean expression for distributive property of OR, AND operators

Distributive property	Boolean expression
OR/Addition	$X \cdot (Y + Z) = X \cdot Y + X \cdot Z$
AND/Multiplication	$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$

The truth table for distributive property of 2-input AND and OR operator is given in Table 4.6.

Table 4.6 Truth table

X	Y	Z	X · Y	X · Z	Y + Z	X · (Y + Z)	(X · Y) + (X · Z)	Y · Z	X + (Y · Z)	X + Y	X + Z	(X + Y) · (X + Z)
0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0	0	0	0	1	0
0	1	1	0	0	1	0	0	1	1	1	1	1
1	0	0	0	0	0	0	0	0	1	1	1	1
1	1	0	1	0	1	1	1	0	1	1	1	1
1	0	1	0	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

The above table proves the distributive property.

DE MORGAN'S THEOREM

Augustus De Morgan developed theorems on Boolean algebra. There are two important theorems stated by De Morgan that can be used to simplify the Boolean expression further.

Theorem 1

The complement of two variables ANDed is equivalent to the OR of the complements of the individual variables. The Boolean expression for Theorem 1 can be depicted as in Figure 4.1.

$$\overline{(X \cdot Y)} = \overline{X} + \overline{Y}$$

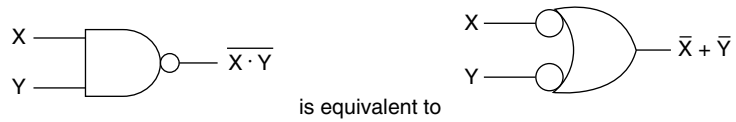


Fig.4.1 Depiction of first De Morgan's theorem

This theorem is verified from the following truth table as in Table 4.7.

Table 4.7 Truth table for theorem 1

X	Y	X′	Y′	XY	(XY)′	X′ + Y′
0	0	1	1	0	1	1
1	0	0	1	0	1	1
0	1	1	0	0	1	1
1	1	0	0	1	0	0

Theorem 2

The complement of two variables ORed is equivalent to the AND of the complements of the individual variables. This theorem can be expressed using the following formula:

$$\left(\overline{X + Y}\right) = \overline{X} \cdot \overline{Y}$$

and can be depicted using logic gate as in Figure 4.2.

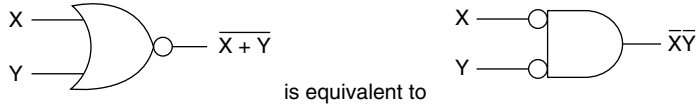


Fig.4.2 Logic representation of theorem 2

The truth table for the second theorem is shown in Table 4.8.

Table 4.8 Truth table for theorem 2

X	Y	X′	Y′	X′Y′	X + Y	(X + Y)′
0	0	1	1	1	0	0
1	0	0	1	0	1	1
0	1	1	0	0	1	1
1	1	0	0	0	1	1

LOGIC AXIOMS

There exist numerous axioms that help in reducing the complexity of a digital circuit. These theorems are broadly classified based on the number of inputs into:

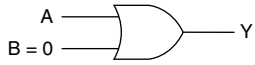
1. Single variant (or) input theorem
2. Multiple variant (or) input theorem

Single Variant Theorems

These theorems are stated below with their corresponding logic diagram and truth table. The logic diagrams and their truth table representing single variant theorem is shown below.

1. A + 0 = A

The truth table proves the axiom.

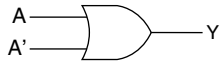


A	B	Y = A + B
0	0	0
1	0	1
0	0	0
1	0	1

2. A . 1 = A

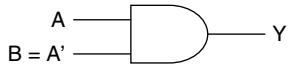
A	B	Y = A · 1
0	1	0
1	1	1
0	1	0
1	1	1

3. A + A' = 1



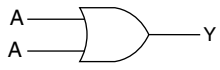
A	A′	Y = A + A′
0	1	1
1	0	1
0	1	1
1	0	1

4. A . A' = 0



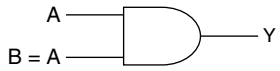
A	A′	Y = A · A′
0	1	0
1	0	0
0	1	0
1	0	0

5. $A + A = A$



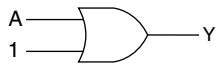
A	A	Y = A + A′
1	1	1
0	0	0
1	1	1
0	0	0

6. $A \cdot A = A$



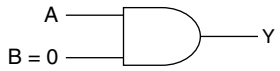
A	A	Y = A · A
0	0	0
1	1	1
0	0	0
1	1	1

7. $A + 1 = 1$



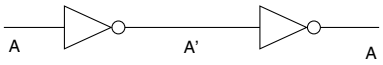
A	1	Y = A + 1
0	1	1
1	1	1
0	1	1
1	1	1

8. $A \cdot 0 = 0$



A	0	Y = A · 0
0	0	0
1	0	0
0	0	0
1	0	0

9. (A')' = A



A	A′	Y = (A′)′
0	1	0
1	0	1
0	1	0
1	0	1

Thus, a close analysis of each of the truth table verifies its corresponding theorem.

Multivariate Theorems

The commutative laws, associative laws, distributive laws, and De Morgan's theorems of Boolean algebra are collectively known as multivariate theorems. These theorems can access more than one variable input.

Rules Considered while Operating Logic Expressions

In any Boolean expression, the order of performing the action is important. The order is as follows:

1. Parenthesis
2. Inversion
3. AND followed by OR
4. Operation of logic gates followed by NOT operation

While performing logical operations, the terms in bracket are to be given first priority. Secondly, the inversion operations have to be performed on individual terms. Thirdly, amongst the basic operators AND is performed first followed by OR operation in any given Boolean expression. Only after all logic operations are completed on the given Boolean expression, NOT operation is performed. For example,

$$A \cdot (A + B)'$$

In the above example, the OR operator in the parenthesis is performed first. First, the inputs A and B are ORed or summed up. The resultant quantity is inversed using NOT operation. Lastly, it is ANDed or multiplied with A to result in the final desired logic expression by its inversion. Then to the outcome AND operation is performed.

MINTERMS AND MAXTERMS

There are number of ways for expressing a Boolean function. One such is using the minterm and maxterm.

Minterm

In general, it is defined as a product of the variables involved in the Boolean function. Each variable in minterm be present only once either in the normal or complemented form. It is denoted as (m_i), where i corresponds the input combination at which the minterm is equal to 1. Consider a 3-input AND gate. The minterms are as represented in Table 4.8.

Table 4.8 Truth table for 3-input AND gate and its minterms

A	B	C	A . B . C	Minterms
0	0	0	0	A′B′C′
0	1	0	0	A′BC′
0	0	1	0	A′B′C
0	1	1	0	A′BC
1	0	0	0	AB′C′
1	1	0	0	ABC′
1	0	1	0	AB′C
1	1	1	1	ABC

It is evident from the above truth table there are 8 combinations or 2^3 combinations accessible for a 3-input AND gate. These are the minterms possible for three inputs AND gate. Therefore, in general, for n inputs there are 2^n probable minterms.

Maxterms

Maxterms are defined as the sum of the variables involved in the Boolean function. Each variable in maxterm be present only once either in the normal or complemented form. It is denoted as (M_i), where i corresponds the input combination at which the maxterm is equal to 0.

Consider the 2-input OR gate. Let A be 1 and A' be 0 similarly, B be 1 and B' be 0. The four combinations of the output is shown in the following truth Table 4.9.

Table 4.9 Truth table for 2-input OR gate and its maxterms

A	B	A + B	Maxterms
0	0	0	A′B′
1	0	1	AB′
0	1	1	A′B
1	1	1	AB

These four combinations are called maxterms. Therefore, for 2-input OR gate there are 4 combinations or 2^2 maxterms.

Similarly, consider a 3-input OR gate's truth table is shown in Table 4.10.

Table 4.10 Truth table for 3-input OR gate and its maxterms

A	B	C	A + B + C	Maxterms
0	0	0	0	A′B′C′
0	1	0	1	A′BC′
0	0	1	1	A′B′C
0	1	1	1	A′BC
1	0	0	1	AB′C′
1	1	0	1	ABC′
1	0	1	1	AB′C
1	1	1	1	ABC

It is evident from the table that there exist 8 combinations or 2^3 maxterms.
In general,

Minterms for n variables = $m_0, m_{1,} \ldots m_{2^n-1}$

Maxterms for n variables = $M_0, M_{1,} \ldots M_{2^n-1}$

Minterms and maxterms are complement of each other.

$$M_i = m_i$$

where $i = 0, 1, 2, \ldots (2^n - 1)$.

The sum of minterms and the product of maxterms forms are known as the canonical forms of a function. Thus, from the minterms and maxterms the canonical form of any Boolean expression can be deduced. These minterms and maxterms are also used in simplifying the Boolean expressions in product of sum and sum of product forms.

ADDITIONAL EXAMPLES

Example 4.1 Prove the identities using Boolean algebra:

1. $\overline{X + Y} = \overline{X} \cdot \overline{Y}$
2. $\overline{X \cdot Y} = \overline{X} + \overline{Y}$

Solution:

1. $\overline{X + Y} = \overline{X} \cdot \overline{Y}$

***Proof*:**

L.H.S.

We know that $X + \overline{X} = 1$ and $X \cdot \overline{X} = 0$ then if we prove $(X + Y) + (\overline{X} \cdot \overline{Y}) = 1$ and $(X + Y) \cdot (\overline{X}\overline{Y}) = 0$. 1st theorem is proved.

$$\begin{aligned}(X+Y)+(\overline{X}\cdot\overline{Y}) &= \left[(X+Y)+\overline{X}\right]\cdot\left[(X+Y)+\overline{Y}\right] \\ &= \left[Y+(X+\overline{X})\right]\cdot\left[X+(Y+\overline{Y})\right] \\ &= (Y+1)\cdot(X+1) \quad \text{since } Y+\overline{Y}=1 \text{ and } X+\overline{X}=1 \\ &= 1\cdot 1 \\ &= 1 \qquad\qquad\qquad\qquad Y+1=1 \text{ and } X+1=1\end{aligned}$$

R.H.S.

$$\begin{aligned}(X+Y)\cdot(\overline{X}\cdot\overline{Y}) &= \left[X\cdot(\overline{X}\cdot\overline{Y})+Y(\overline{X}\cdot\overline{Y})\right] \\ &= \left[(X\cdot\overline{X})\cdot\overline{Y}\right]+\left[(Y\cdot\overline{Y})\cdot\overline{X}\right] \\ &= 0+0 \\ &= 0\end{aligned}$$

2. $\overline{X\cdot Y}=\overline{X}+\overline{Y}$

***Proof*:**

The Boolean law states that $X+\overline{X}=1$ and $\overline{X}\cdot X=0$ i.e., $X\cdot Y+\overline{X\cdot Y}=1$ and $(X\cdot Y)\cdot\overline{XY}=0$. Therefore, the above equation can be proved if we express $X\cdot Y+(\overline{X}+\overline{Y})=1$ and $(X\cdot Y)\cdot(\overline{X}\cdot\overline{Y})=0$

$$\begin{aligned}X\cdot Y+(\overline{X}+\overline{Y}) &= (\overline{X}+\overline{Y})+(X\cdot Y) \\ &= \left[(\overline{X}+\overline{Y})+X\right]\cdot\left[(\overline{X}+\overline{Y}+Y)\right] \\ &= (X+\overline{X}+\overline{Y})\cdot(Y+\overline{Y}+\overline{X}) \\ &= (1+\overline{Y})\cdot(1+\overline{X})=1.1=1\end{aligned}$$

$$\begin{aligned}X\cdot Y(\overline{X}+\overline{Y}) &= \left[(X\cdot Y)\cdot\overline{X}\right]+\left[(X\cdot Y)\cdot\overline{Y}\right] \\ &= \left[(X\cdot\overline{X})\cdot Y\right]+\left[X\cdot(Y\cdot\overline{Y})\right] \\ &= (0\cdot Y)+(0\cdot X) \\ &= 0+0 \\ &= 0\end{aligned}$$

Hence proved.

Example 4.2 Prove the identities using Boolean algebra:

1. $\overline{A}BC+\overline{A}\overline{B}C+A\overline{B}=AC+\overline{AB}$
2. $\overline{A}(A+C)(AB+\overline{C})=0$

Solution:

1. $\overline{A}BC + \overline{A}\overline{B}C + A\overline{B} = AC + \overline{AB}$

$$\begin{aligned} \text{L.H.S.} &= \overline{A}BC + \overline{A}\overline{B}C + A\overline{B} \\ &= \overline{A}C(B + \overline{B}) + \overline{AB} \qquad \text{since } B + \overline{B} = 1 \\ &= \overline{A}C + \overline{AB} = \text{R.H.S} \end{aligned}$$

2. $\overline{A}(A + C)(AB + \overline{C}) = 0$

$$\begin{aligned} \text{L.H.S.} &= \overline{A}(A + C)(AB + \overline{C}) \\ &= (\overline{A}A + \overline{A}C)(AB + \overline{C}) \\ &= (0 + \overline{A}C)(AB + \overline{C}) \qquad \overline{A}A = 0 \\ &= \overline{A}C(AB + \overline{C}) \\ &= \overline{A}ABC + \overline{A}C\overline{C} \qquad \text{since } \overline{A}A = 0,\ C\overline{C} = 0 \\ &= 0 \cdot BC + \overline{A} \cdot 0 \\ &= 0 \end{aligned}$$

Example 4.3 Prove the identity using Boolean algebra and De Morgan's theorems:

1. $\overline{A + B + \overline{A}\overline{B} + AB} = \overline{A} \cdot \overline{B}$

2. $ABC + \overline{A}BC + A\overline{B}C + \overline{A}B\overline{C} + \overline{A}\overline{B}\overline{C} = (A + B)C + \overline{A}\overline{C}$

Solution:

1. $\overline{A + B + \overline{A}\overline{B} + AB} = \overline{A} \cdot \overline{B}$

$$\begin{aligned} \text{L.H.S} &= \overline{A + B + \overline{A}\overline{B} + AB} \\ &= \overline{A + B + \overline{A} + \overline{B} + AB} \quad \text{since } A + \overline{A} = 1 \\ &= \overline{AB} \qquad\qquad\qquad\qquad B + \overline{B} = 1 \\ &= \overline{A \cdot B} \end{aligned}$$

2. $ABC + \bar{A}BC + A\bar{B}C + \bar{A}B\bar{C} + \bar{A}\bar{B}\bar{C} = (A + B)C + \bar{A}\bar{C}$

$$\begin{aligned} \text{L.H.S} &= ABC + \bar{A}BC + A\bar{B}C + \bar{A}B\bar{C} + \bar{A}\bar{B}\bar{C} \\ &= BC(A + \bar{A}) + A\bar{B}C + \bar{A}\bar{C}(B + \bar{B}) \\ &= BC + A\bar{B}C + \bar{A}\bar{C} \\ &= C(B + A\bar{B}) + \bar{A}\bar{C} \\ &= (A + B)C + \bar{A}\bar{C} \end{aligned}$$

Example 4.4 Reduce the given Boolean expression:

$$Y = \bar{A}\bar{B}\bar{C} + AB\bar{C} + A\bar{B}C + ABC + \bar{A}BC$$

Solution: $Y = \bar{A}\bar{B}\bar{C} + AB\bar{C} + A\bar{B}\bar{C} + ABC + \bar{A}BC$

Rearranging the terms:

$$\begin{aligned} &= \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC + \bar{A}BC \\ &= \bar{B}\bar{C}(\bar{A} + A) + AB\bar{C} + BC(A + \bar{A}) \quad \text{since } A + \bar{A} = 1 \\ &= \bar{B}\bar{C} + AB\bar{C} + BC \\ &= \bar{C}(\bar{B} + AB) + BC \\ &= \bar{C}(A + B) + BC \\ &= (A + B)\bar{C} + BC \end{aligned}$$

Hence the given Boolean expression is reduced to $(A + B)\bar{C} + BC$.

Example 4.5 Simplify $AB + A\bar{B}C(\bar{B}\bar{C} + C) + \overline{AC}$.

Solution:

$$\begin{aligned} &= AB + A\bar{B}C(\bar{B}\bar{C} + C) + \overline{AC} \\ &= AB + A\bar{B}\bar{B}C\bar{C} + A\bar{B}CC + \overline{AC} \quad \text{since } C\bar{C} = 0 \\ &= AB + 0 + A\bar{B}C + \overline{AC} \qquad\qquad CC = C \\ &= AB + A\bar{B}C + \bar{A} + \bar{C} \\ &= AB + (A + \bar{A}) \cdot (\bar{A} + \bar{B}C) + \bar{C} \\ &= \bar{C} + AB + 1 \cdot (\bar{A} + \bar{B}C) \\ &= \bar{A} + A \cdot B + \bar{C} + \bar{B} \cdot C \\ &= (\bar{A} + A) \cdot (\bar{A} + B) + (\bar{C} + \bar{B}) \cdot (\bar{C} + C) \\ &= 1 \cdot (\bar{A} + B) + (\bar{C} + \bar{B}) \cdot 1 \\ &= \bar{A} + B + \bar{C} + \bar{B} \end{aligned}$$

Thus, simplified.

Example 4.6 Using Boolean algebra simplify the expression:

$$Y = AC + A(B + C) + C(B + C)$$

Solution: $Y = AC + A(B + C) + C(B + C)$

By applying distributive law to the second and third terms we get:

$$\begin{aligned}
&= AC + AB + AC + BC + CC \\
&= AC + AB + AC + BC + C \quad \text{since } CC = C \\
&= AC + AC + AB + BC + C \qquad AC + AC = AC \\
&= AC + AB + BC + C \\
&= AC + AB + C \qquad \text{since } BC + C = C \\
&= AB + AC + C \qquad AC + C = C \\
&= AB + C
\end{aligned}$$

The expression is simplified.

Example 4.7 Solve using Boolean laws:

1. $(A + \overline{B}C + B)\overline{B}$
2. $A \cdot B + \overline{A} \cdot \overline{B} + A \cdot C$

Solution:

1. $(A + \overline{B}C + B)\overline{B}$

$$\begin{aligned}
Y &= (A + \overline{B}C + B)\overline{B} \\
&= A\overline{B} + \overline{B}\overline{B}C + B\overline{B} \\
&= A\overline{B} + C\overline{B} + 0 \\
&= \overline{B}(A + C)
\end{aligned}$$

2. $A \cdot B + \overline{A} \cdot \overline{B} + A \cdot C$

$$\begin{aligned}
Y &= A \cdot B + \overline{A} \cdot \overline{B} + A \cdot C \\
&= A \cdot B(C + \overline{C}) + \overline{A}\overline{B}(C + \overline{C}) + AC(B + \overline{B}) \\
&= ABC + AB\overline{C} + \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} + ABC + A\overline{B}C
\end{aligned}$$

Since ABC repeats twice, it is retained only once.

$$= ABC + AB\overline{C} + \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} + A\overline{B}C$$

Hence solved.

Example 4.8 Reduce the Boolean expression:

$$ABC + A\overline{B}C + \overline{A}\overline{B}\overline{C} + AB\overline{C}$$

Solution:

$$\begin{aligned} Y &= ABC + A\overline{B}C + \overline{A}\overline{B}\overline{C} + AB\overline{C} \\ &= AC\left(B + \overline{B}\right) + \overline{C}\left(\overline{A}\overline{B} + AB\right) \\ &= AC(1) + \overline{C}(1) \\ &= AC + \overline{C} \\ &= A \end{aligned}$$

Example 4.9 Reduce the Boolean expression:

$$Y = ABCD + ABC\overline{D} + A\overline{B}CD + \overline{A}\overline{B}C\overline{D}$$

Solution:

$$\begin{aligned} Y &= ABCD + ABC\overline{D} + A\overline{B}CD + A\overline{B}C\overline{D} \\ &= ABC\left(D + \overline{D}\right) + A\overline{B}C\left(D + \overline{D}\right) \\ &= ABC(1) + A\overline{B}C(1) \\ &= ABC + A\overline{B}C \\ &= AC\left(B + \overline{B}\right) \\ &= AC \end{aligned}$$

Example 4.10 Minimize the Boolean expression:

1. $Y = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}\overline{C} + A\overline{B}C$
2. $\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + A\overline{B}CD + A\overline{B}CD + ABCD$

Solution:

1.

$$\begin{aligned} Y &= \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}\overline{C} + A\overline{B}C \\ &= \overline{A}\overline{B}\left(\overline{C} + C\right) + A\overline{B}\left(\overline{C} + C\right) \\ &= \overline{A}\overline{B}(1) + A\overline{B}(1) \\ &= \overline{A}\overline{B} + A\overline{B} \\ &= \overline{B}\left(\overline{A} + A\right) \\ &= \overline{B} \cdot 1 \\ &= \overline{B} \end{aligned}$$

Hence the given expression is reduced.

2. $$\begin{aligned} Y &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + A\bar{B}CD + A\bar{B}CD + ABCD \\ &= \bar{A}\bar{B}\bar{C}(\bar{D} + D) + \bar{B}CD(\bar{A} + A) + ACD(\bar{B} + B) \\ &= \bar{A}\bar{B}\bar{C}(1) + \bar{B}CD(1) + ACD(1) \\ &= \bar{A}\bar{B}\bar{C} + \bar{B}CD + ACD \\ &= \bar{A}\bar{B}\bar{C} + CD(\bar{B} + A) \end{aligned}$$

The expression is reduced at this point.

Example 4.11 Reduce the Boolean expression:

$$ABCD + ABC\bar{D} + AB\bar{C}D + AB\bar{C}\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + A\bar{B}\bar{C}D + \bar{A}B\bar{C}D + \bar{A}\bar{B}\bar{C}D$$

Solution:

$$\begin{aligned} Y &= ABCD + ABC\bar{D} + AB\bar{C}D + AB\bar{C}\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + A\bar{B}\bar{C}D + \bar{A}B\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} \\ &= ABC(D + \bar{D}) + AB\bar{C}(D + \bar{D}) + A\bar{B}C(D + \bar{D}) + \bar{B}\bar{C}(AD + \bar{A}\bar{D}) + \bar{A}B\bar{C}D \\ &= ABC(1) + AB\bar{C}(1) + A\bar{B}C(1) + \bar{B}\bar{C}(1) + \bar{A}B\bar{C}D \\ &= ABC + AB\bar{C} + A\bar{B}C + \bar{B}\bar{C} + \bar{A}B\bar{C}D \qquad \text{since } D + \bar{D} = 1 \\ &= AB(C + \bar{C}) + A\bar{B}C + \bar{B}\bar{C} + \bar{A}B\bar{C}D \qquad C + \bar{C} = 1 \\ &= AB + A\bar{B}C + \bar{B}\bar{C} + \bar{A}B\bar{C}D \\ &= A(B + \bar{B}C) + \bar{B}\bar{C} + \bar{A}B\bar{C}D \\ &= AC + \bar{B}\bar{C} + \bar{A}B\bar{C}D \\ &= AC + \bar{C}(B + \bar{A}BD) \end{aligned}$$

Hence reduced.

Example 4.12 Find the Boolean expression for the function f(A, B, C) represented by the table:

A	B	C	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Solution: The terms of the function whose value is 1 in the table is taken and the sum of these terms gives the desired Boolean expression.

$$\overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}C + AB\overline{C}$$

The solution obtained is also called sum of products expansion.

MULTIPLE CHOICE QUESTIONS

1. The first law of Boolean theorem states
   (a) $\overline{A+B} = \overline{A} \cdot \overline{B}$ (b) $\overline{A+B} = \overline{A \cdot B}$
   (c) $\overline{A+B} = \overline{A+B}$ (d) $\overline{A+B} = \overline{A} + \overline{B}$
2. The second law of Boolean theorem states
   (a) $\overline{A \cdot B} = A + B$ (b) $\overline{A \cdot B} = \overline{A} \cdot \overline{B}$
   (c) $\overline{A} \cdot \overline{B} = \overline{A} + \overline{B}$ (d) $\overline{A} \cdot \overline{B} = A + B$
3. Which of the following logical operations is represented by the + sign in Boolean algebra?
   (a) Inversion (b) AND
   (c) OR (d) Complement
4. The format used to present the logic output for the various combinations of logic inputs to a gate is called
   (a) Boolean constant (b) Boolean variable
   (c) Input logic function (d) Truth table
5. Applying De Morgan's theorem to the expression (XYZ)', we get
   (a) (XY)' + Z' (b) X' + Y' + Z'
   (c) X + Y + Z (d) X' · Y' · Z'
6. The methodical reduction of logic circuits is accomplished by using
   (a) Boolean algebra (b) Symbolic reduction
   (c) Transistor – Transistor logic (d) Truth table
7. Which of the examples below expresses the distributive law of Boolean algebra?
   (a) (X + Y) + Z = X + (Y + Z) (b) X + (Y + Z) = XY + XZ
   (c) X(YZ) = (XY) + Z (d) X(Y + Z) = XY + XZ
8. The expression A + A' produces
   (a) 1 (b) 0
   (c) A (d) A'
9. Minterm is the _____ of the variables involved in the Boolean function.
   (a) Sum (b) Difference
   (c) Product (d) None of the above

10. The expression A · A' produces
    (a) 1 (b) 0
    (c) A (d) A'
11. In the Boolean function, maxterms are the _____ of the variables.
    (a) Sum (b) Product
    (c) Difference (d) None of the above
12. A Boolean expression with '*n*' inputs, the probable minterms are
    (a) $2n$ (b) $n - 1$
    (c) $n + 1$ (d) 2^n
13. The expression X + Y = Y + X is an example for
    (a) Commutative law (b) Complement law
    (c) Closure law (d) Distributive law
14. The given expression X + Y is an example of
    (a) Commutative Law (b) Identity
    (c) Closure property (d) None of the above
15. The terms in Boolean expressions that result in '0' are called as
    (a) Minterms (b) Words
    (c) Maxterms (d) Characters
16. A three variable Boolean expression has _____ minterms.
    (a) 0 (b) 2
    (c) 4 (d) 8

Answers

1. (a) 2. (c) 3. (c) 4. (d) 5. (b) 6. (a) 7. (d) 8. (a)
9. (c) 10. (b) 11. (a) 12. (d) 13. (a) 14. (c) 15. (c) 16. (d)

DESCRIPTIVE QUESTIONS

1. Explain and prove the De Morgan's laws.
2. Verify the single input logic axioms with truth table and circuit diagrams.
3. Discuss the closure, associative, commutative and distributive properties of Boolean algebra.
4. What are minterms and maxterms of a logic system?

Simplify the Boolean expression using Boolean postulates

5. $X = C(B + C)(A + B + C)$
6. $Y = \overline{ABC} + \overline{A}B\overline{C} + A\overline{BC} + \overline{ABC}$
7. $\overline{\overline{XY + XYZ} + Y(X + \overline{Y})}$

8. $(\overline{A}+C)(A+\overline{B})(B+C)$
9. $AB+\overline{AB}+AB\overline{C}(AC+B)$
10. Verify that the NAND and NOR operations are commutative but not associative.

Prove the following Boolean algebraic theorems

11. $A+\overline{A}\cdot B+A\cdot\overline{B}=A+B$
12. $\overline{A}BC+A\overline{B}C+AB\overline{C}+ABC=AB+BC+AC$
13. $(A+B)(A+\overline{AB})C+\overline{A}(B+\overline{C})+\overline{A}B+ABC=C(A+B)+\overline{A}(B+\overline{C})$
14. $\overline{A(\overline{A\cdot B})}\cdot\overline{B(\overline{A\cdot B})}=A\oplus B$

KARNAUGH MAPPING

Karnaugh map is a visual depiction of a Boolean expression. This tool was introduced in 1953 by Maurice Karnaugh. It is a scheme of reducing the intricacy and expressing the truth table of a given Boolean expression in the form of a map. The Karnaugh map comprises of small squares put together to form a table. Each cell of the table has one minterm derived from the Boolean expression. A 2-variable Boolean expression has four minterms and thus, has four cells in the Karnaugh map. In general, n variables have 2^n cells in the Karnaugh map. While performing the sum of the product method "1" is taken as the variable and "0" as its complement. Thus, a 2-variable map has 4 squares, a 3-variable map has 8 squares, and a 4-variable map has 16 squares and so on. Before working on the Karnaugh map of the given table, one must know few terminologies used in Boolean logic. Implicant is a term derived by the sum or product of two minterms. Implicants are usually enclosed by a rectangular blocks in a Karnaugh map. The combination of minterms leads to simplified terms of a given K-map. Implicants are of two types:

1. **Prime implicants:** These are implicants which do not form sub set of other implicants in the map. Each product term implicant is individual. The prime implicants cannot be reduced or simplified any further.
2. **Essential prime implicants:** A prime implicant representing "1" in the map if is not a part of other prime implicants of the K-map and cannot be ignored is a essential prime implicant.

The Karnaugh mapping tool can be performed for a maximum of six variables but is best for three and four variable system. In a Karnaugh map, the Boolean variables are transferred (generally from a truth table) and ordered according to the principles of gray code in which only one variable changes in between squares. Once the table is generated and the output possibilities are transcribed, the data is arranged into the largest even group possible and the minterm is generated through the axiom laws of Boolean algebra.

KARNAUGH MAP FOR TWO VARIABLES

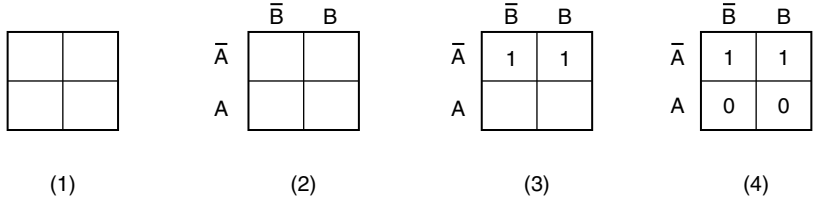
Consider a 2-variable Boolean expression having the truth table as shown in Figure 5.1.

A	B	Y
0	1	1
1	1	0
0	0	1
1	0	0

Fig.5.1

The Karnaugh map for the given truth table is drawn in the following steps:

1. A table comprising of four cells are drawn.
2. The complement of the variables are first written followed by the uncomplemented variables.
3. The 1's from the table is filled in the map.
4. Finally, the 0's are filled.



Following are the Karnaugh map for 2-input OR and AND gates, respectively.

K-map for 2-variable OR gate K-map for 2-variable AND gate

Any given Boolean expression can be depicted as a Karnaugh map in the following formats:

- **Sum of the product form (SOP):** In the sum of the product term, the product of the minterms abiding with the given Boolean expression are taken and then they are summed up together. E.g.,

$$AB + BD$$

- **Product of the sum form (POS):** In product of sum term, the sum of the minterms is performed first and later their product is taken. E.g.,

$$(A + B)\,(B + D)$$

Both of these methods are implemented to simplify the complexicity of the Boolean expression.

KARNAUGH MAP FOR THREE VARIABLES

Karnaugh map is best proven for 3-variable Boolean expressions since they minimize the complexity to greater extent. As discussed earlier, a 3-variable input has eight cells in the Karnaugh map. A 3-variable Boolean expression having the truth table as shown in Figure 5.2 is considered.

A	B	C
0	0	0
0	1	0
0	0	1
0	1	1
1	0	0
1	1	0
1	0	1
1	1	1

Fig.5.2

The Karnaugh map for the given truth table is drawn in the following steps:

1. A table comprising of eight cells are drawn.
2. The complement of the variables are first written followed by the uncomplemented variables.
3. The 1's from the table is filled in the map.
4. Finally, the 0's are filled.

	$\overline{C}$	C
$\overline{A}\,\overline{B}$	0	0
$\overline{A}\,B$	0	0
$A\,\overline{B}$	0	1
$A\,B$	0	1

KARNAUGH MAP FOR FOUR VARIABLES

A 4-variable Karnaugh map consists of 16 squares or cells for each minterm. The gray code representation of the map is depicted in Figure 5.3.

In the map, m_7 is derived from (10) and (11) which gives the binary code (1011) whose decimal equivalent is 7. Similarly, m_{13} is obtained from (11) and (10) that give 13 as its decimal equivalent. Thus, the map is filled with the gray code equivalents.

	$\bar{C}\bar{D}$	$C\bar{D}$	CD	$\bar{C}D$
$\bar{A}\bar{B}$	m_0	m_1	m_3	m_2
$A\bar{B}$	m_4	m_5	m_7	m_6
AB	m_{12}	m_{13}	m_{15}	m_{14}
$\bar{A}B$	m_8	m_9	m_{11}	m_{10}

Fig.5.3

KARNAUGH MAP FOR FIVE VARIABLES

A Karnaugh map for 5-variable expression can be denoted with two 4-variable maps one beside the other. This is depicted as shown in Figure 5.4. The map has 32 cells to fill each minterm. As the number of variables keeps increasing, the efficacy of Karnaugh map decreases. The five variables are A, B, C, D and E. Variable E has two conditions. For the minterms 0 to 15, E is 0 and for minterms 16 to 31 variable E takes the value 1.

E = 0

	$\bar{C}\bar{D}$	$C\bar{D}$	$\bar{C}D$	CD
$\bar{A}\bar{B}$	m_0	m_1	m_2	m_3
$A\bar{B}$	m_4	m_5	m_6	m_7
$\bar{A}B$	m_8	m_9	m_{10}	m_{11}
AB	m_{12}	m_{13}	m_{14}	m_{15}

E = 1

	$\bar{C}\bar{D}$	$C\bar{D}$	$\bar{C}D$	CD
$\bar{A}\bar{B}$	m_{16}	m_{17}	m_{18}	m_{19}
$A\bar{B}$	m_{20}	m_{21}	m_{22}	m_{23}
AB	m_{24}	m_{25}	m_{26}	m_{27}
$\bar{A}B$	m_{28}	m_{29}	m_{30}	m_{31}

Fig.5.4

It is to be noted only one bit changes between the adjacent cells in both the map. This can be achieved only by representing the map in gray code. While forming groups both the maps are visualized one on top of the other. The minterms m_8 and m_{24} are adjacent while m_2 is adjacent to $m_{18.}$ Thus, the minterms of both the maps can be grouped together. Grouping in vertical direction is also allowable in the same approach. But the above procedure proves to be tedious especially for six variables where four maps each consisting of four variables are placed one beside the other. Solving a Karnaugh map for a five and six variables are highly cumbersome and can be more practically solved using computer programs.

Grouping of the bits in a Karnaugh map is done by using following methods.

Pairs

A Karnaugh map always consists of pair of 1's adjacent to each other. This can be well explained from the following example. The steps for obtaining the Karnaugh map is as similar to the 2-variable Boolean expression.

	$\overline{C}$	C
$\overline{A}\,\overline{B}$	0	0
$\overline{A}\,B$	0	0
$A\,\overline{B}$	0	1
$A\,B$	0	1

Notice the first "1" is obtained by the product of $A\overline{B}C$ and the second "1" occurs from the product ABC. Since the 1's are adjacent to each other they form a pair. When we move from the first "1" to the second "1", only one variable changes from the complemented state to the non-complemented state which is B. Thus, the output for these two pairs is written as:

$$\begin{aligned} Y &= A\overline{B}C + ABC \\ &= AC(\overline{B} + B) \qquad (\overline{B} + B = 1) \\ &= AC \end{aligned}$$

A pair is thus obtained by dropping the OR function of a variable and its complement. For simplicity in identifying the pairs they are encircled, so that by seeing the map one can identify that its variable and complement is dropped. The final encircled pair represents the single product.

Second example is that of a 4-variable Boolean expression. Consider the first pair; the first "1" which is obtained from the product of ABCD and the second "1" is from the product ABCD.

The Boolean expression for this is given as:

$$Y = \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD$$

Only D undergoes change from the complemented state to the uncomplemented state and therefore the equation becomes,

$$\begin{aligned} Y_1 &= \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD \\ &= \overline{A}\,\overline{B}C(\overline{D} + D) \\ &= \overline{A}\,\overline{B}C \qquad (\text{since } \overline{D} + D = 1) \end{aligned}$$

In the second pair, the first "1" is from the product $A\overline{B}CD$ and the second "1" is by the product of ABCD. Here the variable B changes from complement to uncomplemented form.

The Boolean expression is given as:

$$\begin{aligned} Y_2 &= AB\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}\,\overline{D} \\ &= A\overline{C}\,\overline{D}(B + \overline{B}) \\ &= A\overline{C}\,\overline{D} \end{aligned}$$

The final Boolean expression is derived by summing both the individual expressions.

$$\begin{aligned} Y &= Y_1 + Y_2 \\ &= \overline{A}\overline{B}C + A\overline{C}\overline{D} \end{aligned}$$

Though "A" appears in the both complement and uncomplemented forms it cannot be neglected. From the above examples it is noticed that a pair will always drop one variable. A 3-variable expression becomes a 2-variable and 4-variable becomes 3-variable.

Quads

In a given Karnaugh map apart from pairs there is possibility of four 1's adjacent to each other forming a quad. The placement of 1's in the Karnaugh map can be of any pattern but comprising of four 1s. Quad formation in a Karnaugh map simplifies the Boolean expression to a much greater extent. A quad eliminates two variables and its complements. Consider the illustration Figure 5.5 of a 4-variable Boolean operation.

	$\overline{C}\overline{D}$	$\overline{C}D$	$C\overline{D}$	CD
$\overline{A}\overline{B}$	0	0	0	1
$\overline{A}B$	0	0	0	1
$A\overline{B}$	0	0	0	1
AB	0	0	0	1

	$\overline{C}\overline{D}$	$\overline{C}D$	$C\overline{D}$	CD
$\overline{A}\overline{B}$	0	0	0	1
$\overline{A}B$	0	0	0	1
$A\overline{B}$	0	0	0	1
AB	0	0	0	1

Fig.5.5

The first 1 in the first pair is obtained by the product of $\overline{A}\,\overline{B}CD$ and $\overline{A}BCD$. The Boolean expression is given by,

$$\begin{aligned} Y_1 &= \overline{A}CD(\overline{B} + B) \\ &= \overline{A}CD \end{aligned}$$

The second pair is by the product of $A\overline{B}CD$ and $ABCD$ for which the Boolean expression is given by,

$$\begin{aligned} Y_2 &= ACD(\overline{B} + B) \\ &= ACD \end{aligned}$$

The final Boolean expression is given as:

$$\begin{aligned} Y &= \overline{A}CD + ACD \\ &= CD(\overline{A} + A) \\ &= CD \end{aligned}$$

Thus, two variables and their complements are eliminated by forming a quad thereby simplifying the Boolean expression.

Octet

A group of eight 1's adjacent to each other is called as an octet. The octet eliminates three variables and its complements.

Consider the following Karnaugh map for four variables. There exist two quads in the map.

	$\overline{C}\overline{D}$	$C\overline{D}$	$\overline{C}D$	CD
$\overline{A}\overline{B}$	1	1	1	1
$A\overline{B}$	1	1	1	1
$\overline{A}B$	0	0	0	0
AB	0	0	0	0

1. The first quad is acquired from the product of,

$$Y_1 = (\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D})(A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D})$$

Now $(\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D}) = \overline{A}\overline{B}\overline{D}$ and $(A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D}) = A\overline{B}\overline{D}$

Therefore, the final product is:

$$Y_1 = (\overline{A}\overline{B}\overline{D}).(A\overline{B}\overline{D})$$
$$Y_1 = \overline{B}\overline{D} \qquad (\text{since } A.\overline{A} = 1)$$

As only these two variables don't change from one form to another.

2. The second quad is obtained from the product of,

$$Y_2 = (\overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD)(A\overline{B}\overline{C}D + A\overline{B}CD)$$

The sum of the product of the term $(\overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD)$ is $\overline{A}\overline{B}D$ and for the term $(A\overline{B}\overline{C}D + A\overline{B}CD)$ is $A\overline{B}D$.

The final product is given by the sum of the two products.

$$Y = \overline{A}\overline{B}D + A\overline{B}D = \overline{B}D$$

The formation of the octet gives a much simplified equation,

$$Y = \overline{B}\overline{D} + \overline{B}D$$
$$= \overline{B}(\overline{D} + D) \qquad (\text{since } \overline{D} + D = 1)$$
$$= \overline{B}$$

As B is the only variable that does not change within the octet. Thus, an octet eliminates three variables and its complements.

Let us consider a Karnaugh map as depicted in Figure 5.6.

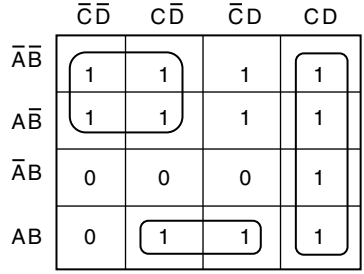


Fig.5.6

Steps to be followed in achieving simplified Boolean expression are as follows:

1. The required cells are drawn and labeled with the variables.
2. On looking at the truth table the 1's are filled in the appropriate cells.
3. Octets are encircled first followed by quad and finally pairs are marked in the map.

Now by implementing pairs, quads and octets the Boolean expression is simplified. Once a Karnaugh map is drawn, the octets are to be encircled first followed by the quad and finally the pairs. Apart from forming octets, quads and pairs there are few more techniques like map rolling, overlapping groups that help in further simplifications of the Boolean expression.

Overlapping Groups

There are cases when the 1's are used in more than one group. One such example is shown in the Karnaugh map in Figure 5.7.

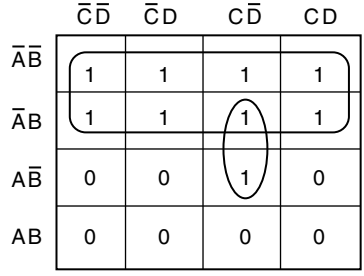


Fig.5.7

There exist an octet and a pair. The "1" obtained by the product of $\overline{A}BC\overline{D}$ is used both in the octet and pair. Such groups are known as overlapping groups or redundant groups. The equation of the above octet is given as:

$$Y = \overline{A} + \overline{A}BC\overline{D}$$

Map Rolling

Another tool used to simplify or to get maximum number of groups is the rolling technique. This technique enables to fold the map imaginarily thus, forming more groups

than the existing ones. In the below example, there exist two pairs at both ends of the Karnaugh map. By rolling the map outward we obtain a quad. As we have already seen that a pair eliminates only one variable and its complement, whereas a quad eradicates two variables and their complements. Rolling is algebraically provable.

The Boolean expression for the above Karnaugh map is obtained by summing the expressions for the individual pairs.

$$\begin{aligned} Y &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} \\ &= \bar{A}\bar{C}\bar{D}(\bar{B} + B) \\ &= \bar{A}\bar{C}\bar{D} \end{aligned}$$

For the second pair, it is

$$\begin{aligned} Y &= \bar{A}\bar{B}C\bar{D} + \bar{A}BC\bar{D} \\ &= \bar{A}C\bar{D}(\bar{B} + B) \\ &= \bar{A}C\bar{D} \end{aligned}$$

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	0	0	1
$\bar{A}B$	1	0	0	1
AB	0	0	0	0
$A\bar{B}$	0	0	0	0

But a quad for the same map reduces the variables which is as follows:

$$\begin{aligned} Y &= \bar{A}\bar{C}\bar{D} + \bar{A}C\bar{D} \\ &= \bar{A}\bar{D}(\bar{C} + C) \\ &= \bar{A}\bar{D} \end{aligned}$$

Thus, rolling the map helps in simplifying the Boolean expression to a greater extent.

Both overlapping and rolling can be performed in a given Karnaugh map to achieve maximum number of groups.

Consider the given map:

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	0	1	1
$\bar{A}B$	1	0	1	1
AB	0	0	1	1
$A\bar{B}$	1	0	1	1

There is an octet and a pair in the Karnaugh map. By rolling and taking into account the overlapping groups, the last "1" can be made to form a pair by rolling from bottom to top. The last step in the Karnaugh mapping is to eliminate the redundant groups.

The equation is given as:

$$Y = \bar{A}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + C$$
$$= \bar{C}(\bar{A} + A\bar{B}) + C$$

The challenge in using Karnaugh maps is selecting the right groups. If the groups are not minimized and if the group size is maximized then the resulting expression will still be equivalent to the original one. This will not be a minimal sum of products. The best approach to find the minimal sum of products (MSP) is to first find all of the largest possible groupings of 1s. These are called prime implicants. The final MSP will contain a subset of these prime implicants.

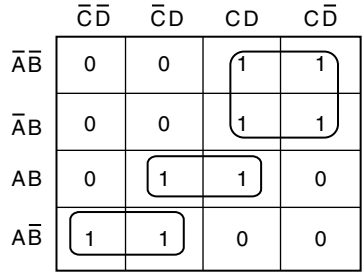
Here is an example of Karnaugh map with prime implicants marked:

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	1
$\bar{A}B$	0	0	1	1
AB	0	1	1	0
$A\bar{B}$	1	1	0	0

The ones in the map can be marked in many ways. Now by pairing the '1's in the following manner:

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	0	0	1	1
$\bar{A}B$	0	0	1	1
AB	0	1	1	0
$A\bar{B}$	1	1	0	0

In the process of forming groups, we sometimes end up in making redundant groups which do not minimize the Boolean expression but rather only equate to the original one. By grouping in the above manner, we end up with many minterms. Thus, it is important to get only the essential prime implicants. The best way of forming the prime implicants is shown below.



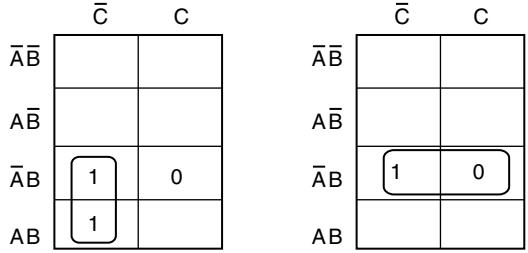
The resulting minimal sum of products is given by the sum of the three groups.

$$Y = \bar{A}C + ABD + A\bar{B}\bar{C}$$

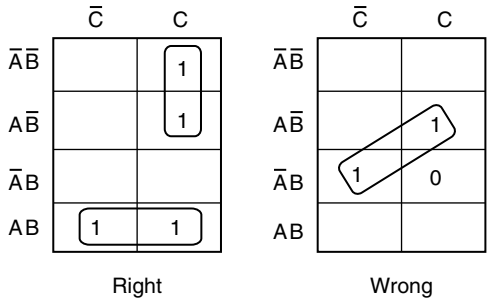
RULES IN KARNAUGH MAPPING

The following rules are incorporated while grouping together closest cells containing '1's for the simplification of expressions.

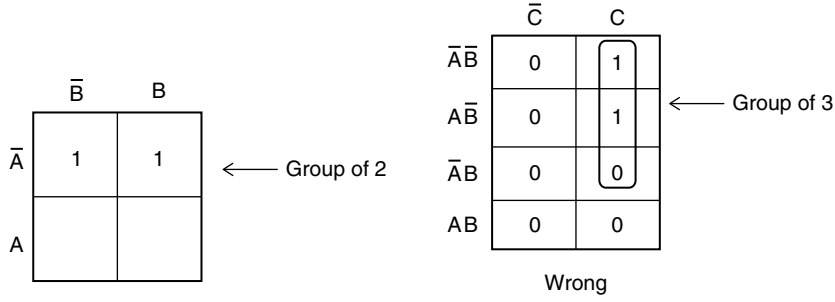
1. Groups formed should not contain a zero.



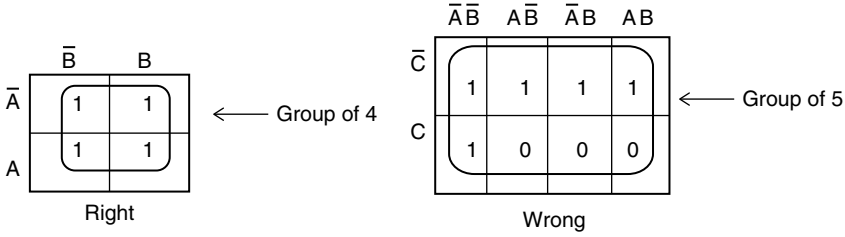
2. Formation of groups may be in vertical or horizontal but NOT diagonal.



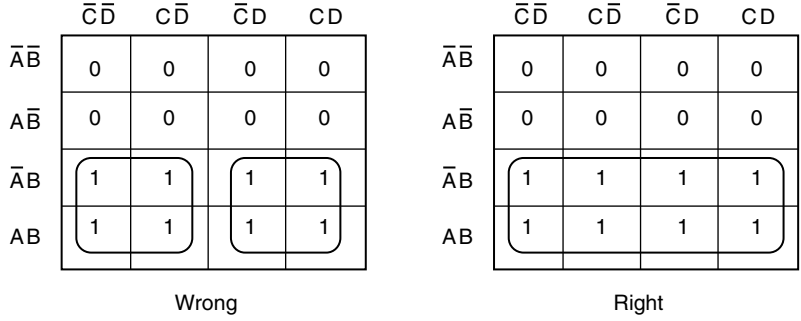
3. The groups in general contain 2^n cells. The number of '1's in the map depends on the n. If $n = 1$, then $2^1 = 2$.



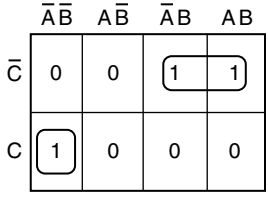
If $n = 2$, the group consists of four 1's since $2^2 = 4$.



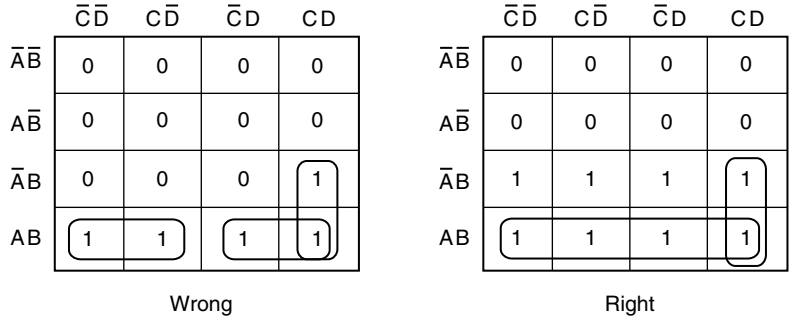
4. The groups must as large as possible without exempting the Boolean laws.



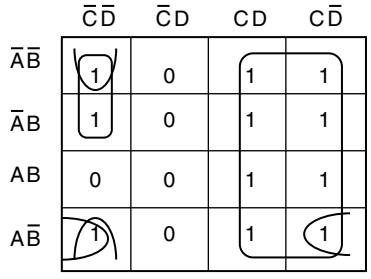
5. The '1's in the Karnaugh map must be grouped in at least one group. If pairing is not possible they can be taken individually but not neglected.



6. Overlapping of groups is allowed to achieve minimal sum of product terms or minimal product of sum terms.

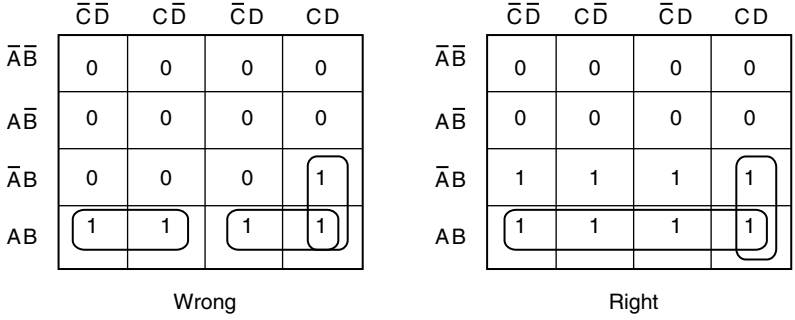

$\bar{C}\bar{D}$ $C\bar{D}$ $\bar{C}D$ CD
$\bar{A}\bar{B}$ 0 0 0 0
$A\bar{B}$ 0 0 0 0
$\bar{A}B$ 0 0 0 1
AB 1 1 1 1
Wrong
$\bar{C}\bar{D}$ $C\bar{D}$ $\bar{C}D$ CD
$\bar{A}\bar{B}$ 0 0 0 0
$A\bar{B}$ 0 0 0 0
$\bar{A}B$ 1 1 1 1
AB 1 1 1 1
Right

7. The map can be rolled in any manner that forms maximum numbers of groups. They can be wrapped from left to right, top to bottom, along the diagonals.



8. The groups must be large at the same time few in number and must abide with the previous rules.

   Example: Where there is a possibility of an octet pairing should be avoided.


$\bar{C}\bar{D}$ $C\bar{D}$ $\bar{C}D$ CD
$\bar{A}\bar{B}$ 0 0 0 0
$A\bar{B}$ 0 0 0 0
$\bar{A}B$ 0 0 0 1
AB 1 1 1 1
Wrong
$\bar{C}\bar{D}$ $C\bar{D}$ $\bar{C}D$ CD
$\bar{A}\bar{B}$ 0 0 0 0
$A\bar{B}$ 0 0 0 0
$\bar{A}B$ 1 1 1 1
AB 1 1 1 1
Right

Product of Sum Form (POS)

Karnaugh mapping is performed with yet another method known as the product of the sum (POS) format. The POS and SOP are similar to each other but with some points of dissimilarity between them.

In the product of the sum form,

$$1 \,.\, 1 = 1 \text{ and } 1.0 = 0.\ 1 = 0$$

The fundamental implicants are procured from the truth table and then these individual terms are ANDed. In POS, the output is 0 for the corresponding input, whereas in SOP the output is 1 for the analogous input.

Transformation of SOP to POS: The product of the sum format can be achieved by converting the sum of the product form by introducing the following steps:

1. Karnaugh map in the SOP form is drawn.
2. The 1's are replaced with 0's and the 0's are replaced with 1's.
3. Octets, quads and pairs are formed by encircling 0's to secure the prime implicants. Redundant groups are eliminated.
4. Simplified equation is written. AND functions are replaced with OR function.
5. Complement of each variable is written.

Example 5.1 Present the POS form of the given Karnaugh map which exist in the sum of the products form.

	$\overline{C}$	C
$\overline{A}\overline{B}$	1	0
$\overline{A}B$	1	0
$A\overline{B}$	0	1
AB	0	1

Solution: Steps to obtain the POS form:

1. The '1' s are replaced with '0's and groups are marked.

	$\overline{C}$	C
$\overline{A}\overline{B}$	0	1
$\overline{A}B$	0	1
$A\overline{B}$	1	0
AB	1	0

2. Equation for the above map is specified as:

$$Y = \overline{A}\,\overline{C} + AC$$

3. The AND function is changed to OR function.

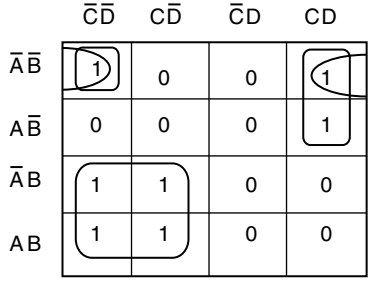
$$Y = (\overline{A} + \overline{C})(A + C)$$

4. Each of the above term is complemented and rewritten as:

$$Y = (A + C)(\overline{A} + \overline{C})$$

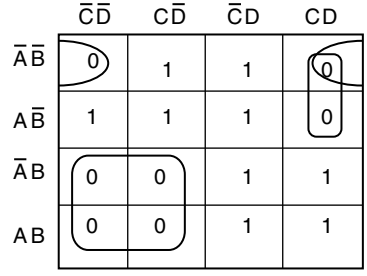
Example 5.2 Express the Boolean expression $Y = \overline{B}CD + B\overline{D} + \overline{A}\overline{B}\overline{C}\overline{D}$ in the POS form.

Solution: The Boolean expression is depicted as a Karnaugh map in the SOP form.



$$= \overline{A}\overline{B}CD + \overline{B}CD + B\overline{D}$$
$$= \overline{B}CD(\overline{A} + 1) + B\overline{D}$$
$$= \overline{B}CD + B\overline{D}$$

The 1's are replaced with 0's and redrawn. Octet, quads and finally pairs are encircled. The redundant groups are eliminated.



The simplified equation is written as:

$$Y = \overline{A}\overline{B}CD + B\overline{D} + \overline{B}CD$$

The AND operators are swapped with OR operators.

$$Y = (\overline{A} + \overline{B} + C + D)(B + \overline{D})(\overline{B} + C + D)$$

Complements of each variable is taken.

$$Y = (A + B + \overline{C} + \overline{D})(\overline{B} + D)(B + \overline{C} + \overline{D})$$

Transformation of POS to SOP: The POS can also be converted into the SOP form. For a given Boolean expression the Karnaugh map is filled with relevant '0's. The 0's are again changed with '1's. Then the groups are formed which leads to the MPS.

Example 5.3 Express the Boolean expression $Y' = (A + B)(\overline{B} + C)$ in SOP form.

Solution: The Karnaugh map for the given map is drawn in the POS form by filling '0's in the relevant cells.

	$\overline{C}$	C
$\overline{A}\overline{B}$	0	0
$A\overline{B}$	1	1
$\overline{A}B$	0	1
AB	0	1

Now the '0's are substituted with '1's in the map and MSP are formed.

	$\overline{C}$	C
$\overline{A}\overline{B}$	1	1
$A\overline{B}$	0	0
$\overline{A}B$	1	0
AB	1	0

Thus, by performing the above steps, product of the sums form is converted to SOP form.

Don't Care Conditions

All sum of the minterms representing a Boolean expression either take the value "1" or "0". There exist certain cases where the conditions are not specified. These don't care conditions can assume any oft the two values. But these don't care conditions can be considered while mapping as it would give a better MSP or MPS. These don't care functions are an outcome of not properly specified minterms. It cannot be assigned "1" or "0" and hence are denoted as "X" in the truth table and map.

While grouping these, 'X's can be considered as either 0 or 1.
Consider one such example of don't care state shown in Figure 5.8.

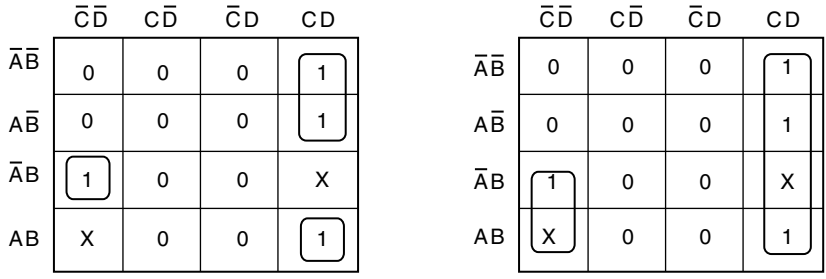

$\bar{C}\bar{D}$ $C\bar{D}$ $\bar{C}D$ CD
$\bar{A}\bar{B}$ 0 0 0 1
$A\bar{B}$ 0 0 0 1
$\bar{A}B$ 1 0 0 X
AB X 0 0 1
$\bar{C}\bar{D}$ $C\bar{D}$ $\bar{C}D$ CD
$\bar{A}\bar{B}$ 0 0 0 1
$A\bar{B}$ 0 0 0 1
$\bar{A}B$ 1 0 0 X
AB X 0 0 1

Fig.5.8

The Boolean expression for the map without considering the don't care conditions is given as:

$$Y = \bar{A}B\bar{C}\bar{D} + \bar{B}CD + ABCD$$

The same map on including the "don't care" conditions and assuming them to take the value "1" results in MSP.

$$Y = \bar{C}D + B\bar{C}\bar{D}$$

ADDITIONAL EXAMPLES

Example 5.4 Find the SOP and POS for the function $f(A, B, C) = \bar{A}B + A\bar{B}$.

Solution: To obtain SOP for the above function we have to multiply the missing variable to both terms.

$$\begin{aligned}
\therefore \quad f(A, B, C) &= \bar{A}B + A\bar{B} \\
&= \bar{A}B(C + \bar{C}) + A\bar{B}(C + \bar{C}) \\
&= \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}C + A\bar{B}\bar{C} \\
&= 011 + 010 + 101 + 100 \\
&= \Sigma(3, 2, 5, 4) \\
&= \Sigma(2, 3, 4, 5)
\end{aligned}$$

$\therefore$ $\bar{A}BC + \bar{A}B\bar{C} + A\bar{B}C + A\bar{B}\bar{C}$ are the required sum of minterms and $\Sigma(2, 3, 4, 5)$ is the canonical SOP form of the function.

To obtain POS form, the missing terms in the function $f(A, B, C)$ is added to both the terms.

$$f(A, B, C) = \bar{A}B + A\bar{B}$$

Before doing this the above expression should be converted to product of sum.

$$\begin{aligned} f(A,B,C) &= \overline{A}B + A\overline{B} + A\overline{A} + B\overline{B} \\ &= \overline{A}B + A\overline{A} + A\overline{B} + B\overline{B} \\ &= \overline{A}(B+A) + \overline{B}(A+B) \\ &= \overline{A}(A+B) + \overline{B}(A+B) \\ &= (\overline{A}+\overline{B})(A+B) \end{aligned}$$

To this the missing variable $C\overline{C}$ is added to both $(A+B)(\overline{A}+\overline{B})$.

$$\begin{aligned} &= (A+B+C\overline{C})(\overline{A}+\overline{B}+C\overline{C}) \\ &= (A+B+C)(A+B+\overline{C})(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C}) \\ &= \quad (000) \qquad (001) \qquad (110) \qquad (111) \\ &= \Pi(0,1,6,7) \end{aligned}$$

$\therefore$ POS = $\Pi(0, 1, 6, 7)$ are the required maxterms or canonical product of sums of the given expression.

Example 5.5 Find the maxterms or canonical POS form of the function $f(A,B,C)$ given in the below table.

Decimal value	A	B	C	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

Solution: The above table gives the decimal values for which the function f assumes the value "0" are (0, 2, 4, 5). Thus by definition, function f(A, B, C) is the product of the sum term is:

$$\begin{aligned} &= \Pi(0,2,4,5,7) \\ &= (000)(010)(100)(101)(111) \end{aligned}$$

$\therefore$ $f(A,B,C) = (A+B+C)(A+\overline{B}+C)(\overline{A}+B+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+\overline{C})$ are the required maxterms or canonical POS for the given function.

Example 5.6 Find the minterms from the given table. Also obtain the simplified function f(A, B, C) in the SOP and POS form. Deduce sum of minterms and product of maxterms.

Decimal value	A	B	C	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

Solution: To obtain the SOP terms, consider the function that assumes 1. i.e., $\Sigma(2,3,6,7)$.

$$= \Sigma(2,3,6,7)$$
$$= 010 + 011 + 110 + 111$$
$$= \overline{A}B\overline{C} + \overline{A}BC + AB\overline{C} + ABC$$

To obtain the POS form, consider the function in table whose value is "0".

$$\Pi(0,1,4,5)$$

$$\therefore \text{ POS} = (000)(001)(100)(101)$$
$$= (A+B+C)(A+B+\overline{C})(\overline{A}+B+C)(\overline{A}+B+\overline{C})$$

Sum of minterms = $\Sigma(2, 3, 6, 7)$

Product of maxterms = $\Pi(0, 1, 4, 5)$

Simplified function in SOP form $= \overline{A}\,\overline{B}\overline{C} + \overline{A}BC + AB\overline{C} + ABC$

Simplified function in POS form $= (A+B+C)(A+B+\overline{C})(\overline{A}+B+C)(\overline{A}+B+\overline{C})$

Example 5.7 Find the sum of minterms for the given function $f(A,B,C,D)$ in the canonical form.

$$f(A,B,C,D) = \overline{A}\,\overline{B}$$

Solution: $f(A,B,C,D) = \overline{A} \cdot \overline{B}$

Since variables C and D are missing therefore by multiplying the terms $(C+\overline{C})$ and $(D+\overline{D})$ we get,

$$f(A,B,C,D) = \overline{A} \cdot \overline{B}(C+\overline{C})(D+\overline{D})$$
$$= (\overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\overline{C})(D+\overline{D})$$
$$= \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\overline{C}D + \overline{A}\,\overline{B}\overline{C}\overline{D}$$
$$= (0011) + (0010) + (0001) + (0000)$$

$$\therefore \text{ SOP form} = \Sigma(3,2,1,0)$$
$$= \Sigma(0,1,2,3)$$

Example 5.8 Find the sum of the product (SOP) and product of the sum (POS) forms for the Boolean expression.

$$f(A,B,C) = \bar{A}B + \bar{B}C + \bar{A}\bar{C}$$

Solution: The SOP form of the expression is given as follows:

$$f(A,B,C) = \bar{A}B + \bar{B}C + \bar{A}\bar{C}$$

To get the SOP form, the missing variable in each term is multiplied to it. Here in the first term $\bar{A}B$ variable "C" is missing and $(C+\bar{C})$ is multiplied to it. The second term does not have the variable A and hence $(A+\bar{A})$ is multiplied to it. In the third term $\bar{A}\bar{C}$ has variable B missing and hence $(B+\bar{B})$ is multiplied.

$$f(A,B,C) = \bar{A}B(C+\bar{C}) + \bar{B}C(A+\bar{A}) + \bar{A}\bar{C}(B+\bar{B})$$
$$= \bar{A}BC + \bar{A}B\bar{C} + \bar{B}CA + \bar{B}C\bar{A} + \bar{A}\bar{C}B + \bar{A}\bar{C}\bar{B}$$
$$= \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}C + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}\bar{C}\bar{B}$$

Retaining once the repeating term $\bar{A}B\bar{C}$ we get,

$$= \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}C + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$$
$$= 011 + 010 + 101 + 001 + 000$$
$$= \Sigma(3,2,5,1,0)$$
$$= \Sigma(0,1,2,3,5)$$

The product of the sum (POS) = Complement of sum of the products (SOP) or complement of $\Sigma(4,6,7)$.

$$= \text{Complement of } \Sigma(100,110,111)$$
$$= \left(A\bar{B}\bar{C} + AB\bar{C} + ABC\right)$$
$$= \overline{A\bar{B}\bar{C} + AB\bar{C} + ABC}$$
$$= \overline{A\bar{B}\bar{C}} \cdot \overline{AB\bar{C}} \cdot \overline{ABC}$$
$$= \left(\bar{A} + \bar{\bar{B}} + \bar{\bar{C}}\right)\left(\bar{A} + \bar{B} + \bar{\bar{C}}\right)(\bar{A} + \bar{B} + \bar{C})$$
$$= (\bar{A} + B + C)(\bar{A} + \bar{B} + C)(\bar{A} + \bar{B} + \bar{C})$$

Thus, the canonical POS form is obtained.

Example 5.9 Find the SOP and POS form for the Boolean expression:

$$f(A,B,C) = \overline{B}\overline{C} + BC + AB + \overline{A}\overline{B}$$

Solution: To determine the SOP form, each of the missing variable is multiplied to its corresponding term $(A + \overline{A})$ is multiplied to $\overline{B}\overline{C}$, $(A + \overline{A})$ to BC also. $(C + \overline{C})$ is multiplied to third term (AB) and 4th term $(\overline{A}\overline{B})$.

$$\begin{aligned} f(A,B,C) &= \overline{B}\overline{C} + BC + AB + \overline{A}\overline{B} \\ &= (A + \overline{A})\overline{B}\overline{C} + (A + \overline{A})BC + AB(C + \overline{C}) + \overline{A}\overline{B}(C + \overline{C}) \\ &= A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C} + ABC + \overline{A}BC + ABC + AB\overline{C} + \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} \end{aligned}$$

The second and third terms are repeating twice. Therefore, retaining these terms only once we get,

$$\begin{aligned} f(A,B,C) &= A\overline{B}\overline{C} + \overline{A}BC + AB\overline{C} + \overline{A}\overline{B}C + ABC + \overline{A}\overline{B}\overline{C} \\ &= 100 + 011 + 110 + 001 + 111 + 000 \\ &= \Sigma(4,3,6,1,7,0) \\ &= \Sigma(0,1,3,4,6,7) \text{ is the SOP form.} \end{aligned}$$

The POS form = complement of SOP.

$$\begin{aligned} \therefore \quad f(A,B,C) &= \Sigma(2,5) \\ &= \Sigma(010,101) \\ &= \overline{A}B\overline{C} + \overline{A}B\overline{C} \\ &= \overline{\overline{A}B\overline{C} + \overline{A}B\overline{C}} \\ &= (\overline{\overline{A}} + \overline{B} + \overline{C})(\overline{\overline{A}} + \overline{B} + \overline{\overline{C}}) \\ &= (A + \overline{B} + \overline{C})(A + \overline{B} + C) \text{ is the POS form.} \end{aligned}$$

Example 5.10 Find the canonical product of sums (POS) and sum of product (SOP) form.

$$f(A,B,C) = A \cdot (\overline{B} + C)$$

Solution: $f(A,B,C) = A \cdot (\overline{B} + C)$

In the first term variables B and C are not present and in the second term variable A is not present. Therefore, adding $B\overline{B} + C\overline{C}$ to the first term and $A\overline{A}$ to the second term we get,

$$\begin{aligned} f(A,B,C) &= (A + B\overline{B} + C\overline{C}) \cdot (\overline{B} + C + A\overline{A}) \\ &= [(A + B)(A + \overline{B}) + C\overline{C}].[(\overline{B} + C + A)(\overline{B} + C + \overline{A})] \end{aligned}$$

$$= \left[C + (A + B)(A + \overline{B})\right]\left[\overline{C} + (A + B)(A + \overline{B})\right]\left[(\overline{B} + C + A)(\overline{B} + C + \overline{A})\right]$$
$$= \left[(C + A + B)\cdot(C + A + \overline{B})\cdot(\overline{C} + A + B)(\overline{C} + A + \overline{B})(\overline{B} + C + A)(\overline{B} + C + \overline{A})\right]$$
$$= \left[(A + B + C)(A + \overline{B} + C)(A + B + \overline{C})(A + \overline{B} + \overline{C})(A + \overline{B} + C)(\overline{A} + \overline{B} + C)\right]$$

Since the second and fifth terms are repeating, it is retained only once.

$$= \left[(A + B + C)(A + \overline{B} + C)(A + B + \overline{C})(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + C)\right]$$
$$= \left[(000)(010)(001)(011)(110)\right]$$
$$= \Pi(0, 2, 1, 3, 6)$$
$$= \Pi(0, 1, 2, 3, 6) \text{ is the required POS form.}$$

To obtain the SOP form for the given function, each of the missing variables in the terms are multiplied.

Therefore,

$$f(A, B, C) = A \cdot (\overline{B} + C)$$
$$= A\overline{B} + AC$$
$$= A\overline{B}(C + \overline{C}) + AC(B + \overline{B})$$
$$= A\overline{B}C + A\overline{B}\overline{C} + ABC + A\overline{B}C$$

The first and last terms are repeated twice. Therefore, retaining it only once we get,

$$= A\overline{B}C + A\overline{B}\overline{C} + ABC$$
$$= 101 + 100 + 111$$
$$= \Sigma(5, 4, 7)$$
$$= \Sigma(4, 5, 7) \text{ is the SOP form}$$

Example 5.11 Find the POS and SOP form for the Boolean expression.

$$f(A, B, C) = A.\overline{B}$$

Solution: The function has three variables A, B and C. The variables C and B are missing in the first term and therefore to obtain POS form we need to add $B\overline{B}$ and $C\overline{C}$ to it. In the second term A and C are missing. Therefore, adding $A\overline{A}$ and $C\overline{C}$ to the second term we get,

$$f(A, B, C) = (A + B\overline{B} + C\overline{C})\cdot(\overline{B} + A\overline{A} + C\overline{C})$$
$$= \left[(A + B)(A + \overline{B})C\overline{C}\right]\cdot\left[(\overline{B} + A)(\overline{B} + \overline{A}) + C\overline{C}\right]$$
$$= \left\{\left[C + (A + B)(A + \overline{B})\right]\left[\overline{C} + (A + B)(A + \overline{B})\right]\right\}$$
$$\left\{\left[C + (\overline{B} + A)(\overline{B} + \overline{A})\right]\left[\overline{C} + (\overline{B} + A)(\overline{B} + \overline{A})\right]\right\}$$

$$= [(C + A + B)(C + A + \overline{B})(\overline{C} + A + B)(\overline{C} + A + \overline{B})(C + \overline{B} + A)$$
$$(C + \overline{B} + \overline{A})(\overline{C} + \overline{B} + A)(\overline{C} + \overline{B} + \overline{A})]$$
$$= [(A + B + C)(A + \overline{B} + C)(A + B + \overline{C})(A + \overline{B} + \overline{C})(A + \overline{B} + C)$$
$$(\overline{A} + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + \overline{C})]$$

The terms $(A + \overline{B} + C)$ is repeated twice, retaining it only once we get,

$$= (A + B + C)(A + \overline{B} + C)(A + B + \overline{C})(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + \overline{C})$$
$$= (000)(010)(001)(011)(110)(011)(111)$$
$$= \prod(0, 2, 1, 3, 7, 6)$$
$$= \prod(0, 2, 1, 3, 7, 6) \text{ is POS form.}$$

To obtain SOP we have to multiply the given Boolean function with the missing term,

$$f(A, B, C) = A \cdot \overline{B}$$
$$= (C + \overline{C})A \cdot \overline{B}$$
$$= A\overline{B}C + A\overline{B}\overline{C}$$
$$= 101 + 100$$
$$= \Sigma(5, 4)$$

SOP form is $\Sigma(4, 5)$.

Example 5.12 Map the following SOP on a Karnaugh map.

$$f(A, B, C) = A + A\overline{B} + \overline{A}B\overline{C}$$

Solution: The SOP expression is not in standard form as each of the terms do not have three variables. The first term has variables B and C missing. In the second term variable C is missing. The terms are expanded numerically as:

A	$+ A\overline{B}$	$+ \overline{A}B\overline{C}$
100	100	010
101	101	
110		
111		

AB \ C	0	1
00		
01	1	
10	1	1
11	1	1

Example 5.13 Map the given SOP on a Karnaugh map.

$$f(A,B,C,D) = \overline{A}\,\overline{C} + C\overline{B} + AB\overline{C} + \overline{A}\,\overline{B}C\overline{D} + \overline{A}BC\overline{D}$$

Solution: The SOP expression is obviously not in standard form as not all terms have all four variables. We shall first expand the terms numerically as:

$\overline{A}\,\overline{C}$	+	$\overline{B}C$	+	$AB\overline{C}$	+	$\overline{A}\,\overline{B}C\overline{D}$	+	$\overline{A}BC\overline{D}$
0000		0010		1100		0010		0110
0001		0011		1101				
1000		1010						
1001		1011						

In the Karnaugh map, the terms that are present are marked a value 1 in the appropriate cell of the 4-variable Karnaugh map.

AB \ CD	00	01	11	10
00	1	1	1	1
01				1
11	1	1		
10	1	1		1

Example 5.14 Use the Karnaugh mapping to minimize the following SOP expression:

$$\overline{A}BC + \overline{A}\,\overline{B}\,\overline{C} + AB\overline{C} + \overline{A}\,\overline{B}C$$

Solution:

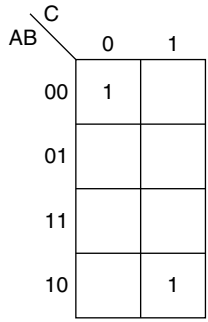
$$\overline{A}BC + \overline{A}\,\overline{B}\,\overline{C} + AB\overline{C} + \overline{A}\,\overline{B}C$$
$$011 + 000 + 110 + 001$$

AB \ C	0	1
00	1	1
01		1
11	1	
10		

After grouping the expression is minimized as:

$$\overline{A}\,\overline{B} + \overline{A}C + AB\overline{C}$$

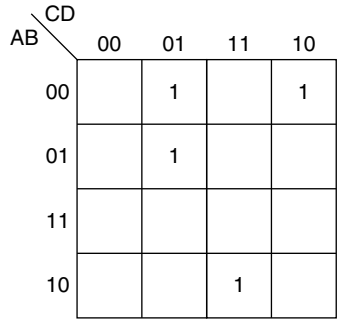
Example 5.15 Determine the SOP for the given K-map.



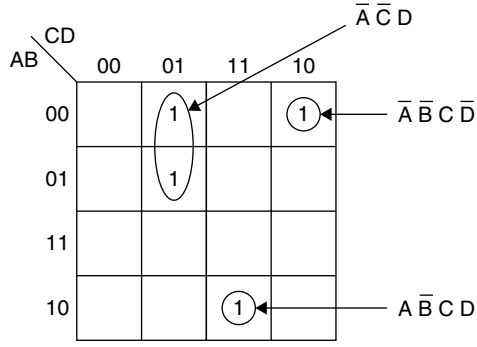
Solution: The SOP expression for the given K-map is given as:

$$\overline{A}\overline{B}\overline{C} + A\overline{B}C$$

Example 5.16 Determine the SOP for the given K-map.



Solution: The given 4-variable K-map is grouped for pairs, quads. The formation of pairs minimizes the SOP. The given K-map is paired as:



The SOP expression = $\overline{A}\overline{C}D + \overline{A}B\overline{C}\overline{D} + AB\overline{C}D$.

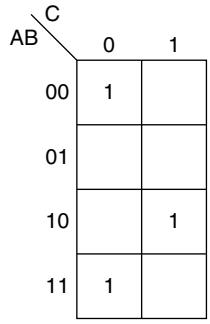
Example 5.17 Map the following POS expression on K-map.

$$(\overline{A}+\overline{B}+\overline{C})(A+B+C)(\overline{A}+B+C)$$

Solution: $(\overline{A}+\overline{B}+\overline{C})(A+B+C)(\overline{A}+B+C)$

The numbers are assigned to the given POS as:

$\overline{A}+\overline{B}+\overline{C}$	$A+B+C$	$\overline{A}+B+C$
111	000	100



Example 5.18 Map the standard POS expression on the K-map. Also minimize the expression.

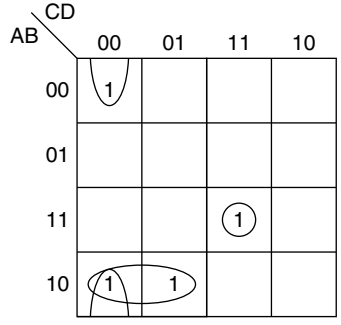
$$(A+B+C+D)(\overline{A}+B+C+D)(\overline{A}+B+C+\overline{D})(\overline{A}+\overline{B}+\overline{C}+D)$$

Solution:

$$(A+B+C+D)\ (\overline{A}+B+C+D)\ (\overline{A}+B+C+\overline{D})\ (\overline{A}+\overline{B}+\overline{C}+\overline{D})$$

0000 1000 1001 1111

The above numbers are assigned value 1 in the appropriate cell of a 4-variable K-map.



The grouping is performed as shown and the minimized expression is given as:

$$\text{Minimized POS} = \overline{A}BC + \overline{A}\overline{B}\overline{C}\overline{D} + BCD$$

SUMMARY

Karnaugh maps are considered as an alternative to algebra for simplifying Boolean expressions. The application of this tool results in minimal sum of products, which in turns minimizes components of a circuit. It is uncomplicated to handle the don't care conditions. This technique holds good for manual simplification of small expressions. While performing mapping, the rules are to be kept in mind.

MULTIPLE CHOICE QUESTIONS

1. A Karnaugh map (K-map) is a theoretical form of representing _____.
   (a) Circuit diagram (b) Block diagram
   (c) Logic diagram (d) Venn diagram
2. Which of the following method is used to minimize Boolean expressions?
   (a) Fourier transform (b) Gray code
   (c) Karnaugh mapping (d) Venitch method
3. Four adjacent '1's in a Karnaugh map forms a
   (a) octet (b) singlet
   (c) pair (d) quad
4. If n denotes the number of variable then the number of cells are given as
   (a) 2^n (b) $2 + n$
   (c) $2 - n$ (d) 2_n
5. A 4-variable Karnaugh map has
   (a) 12 cells (b) 16 cells
   (c) 18 cells (d) 20 cells
6. The Boolean expression Y = XY + ZX is in the _____ form.
   (a) Product-of-Sum (b) Sum-of-Products
   (c) Linear (d) None of the above
7. In the Karnaugh map, each cell represents _____ minterm derived from the Boolean expression.
   (a) 1 (b) 2
   (c) 3 (d) 4
8. Which of the following is NOT considered for forming groups in K-map?
   (a) Rolling (b) Diagonal
   (c) Vertical (d) Horizontal
9. The sum or product of two minterms results in
   (a) Maxterms (b) Boolean expression
   (c) Implicant (d) None of the above

10. While forming groups in K-mapping the Don't care states are combined along with terms of
    (a) Minterms (b) Maxterm
    (c) SOP (d) POS
11. Product-of-Sums expressions can be implemented using
    (a) 2-level OR-AND logic circuits (b) 2-level NOR logic circuits
    (c) 2-level XOR logic circuits (d) Both (a) and (b)
12. In a Karnaugh map, a group of eight 1's adjacent to each other is called
    (a) Pairs (b) Triad
    (c) Quads (d) Octet
13. The Boolean expressions are represented in a unique way called
    (a) Canonical form (b) Minterm
    (c) Maxterm (d) Logic diagram
14. Don't care conditions can be used for simplifying Boolean expression in
    (a) Logic diagram (b) Minterms
    (c) K-maps (d) Maxterms
15. In a Karnaugh map the formation of Quad results in the elimination of _____ variables and their complements.
    (a) 2 (b) 3
    (c) 4 (d) 8
16. A 2-level AND-OR logic circuit is used to express
    (a) Product of sum (b) Sum of product
    (c) Boolean expression (d) All of the above

Answers

1. (d) 2. (c) 3. (d) 4. (a) 5. (b) 6. (b) 7. (a) 8. (b)
9. (c) 10. (a) 11. (d) 12. (d) 13. (a) 14. (c) 15. (a) 16. (b)

DESCRIPTIVE QUESTIONS

1. Explain a 4-variable Karnaugh map.
2. Write short notes on:
   (a) Pairs
   (b) Quad
   (c) Octet
   (d) Don't care conditions
3. Discuss the rules considered while performing Karnaugh map technique.
4. Simplify the Boolean expression $Y = \overline{A}\,\overline{B}\,\overline{C}D + \overline{B}CD + \overline{A}CD + ABC$
5. Transform the Boolean expression $Y = B + A'BC$ into POS form.

6. Simplify the following expression into sum of products using Karnaugh map $f(A, B, C, D) = \sum(1, 3, 4, 5, 6, 7, 9, 12, 13)$
7. Minimize the logic function $Y(A, B, C, D) = \sum(0, 1, 2, 3, 5, 6, 7)$ using Karnaugh map.
8. Find the minterms and SOP form of the function $f(A, B, C)$ whose truth table given below:

A	B	C	f(A, B, C)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

9. Find the maxterms and POS form of the function $f(A, B, C)$ whose truth table given below:

A	B	C	f(A, B, C)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

10. Find the SOP and POS form of the function $f(A, B, C)$ whose truth table given below:

Decimal value	A	B	C	f(A, B, C)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

11. Find the SOP and POS form of the function f(A, B, C) whose truth table given below:

Decimal value	A	B	C	f(A, B, C)
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

12. Find the sum of the minterms and product of the maxterms expressions for the function $f(A, B, C) = A + B\overline{C}$
13. Minimize the following logic function using K-maps and realize using NAND and NOR gates.

$$f(A, B, C, D) = \sum(1, 3, 11, 15) + \Phi(2, 13)$$

14. Obtain the minimal SOP expression for the function given below using Karnaugh map.

$$f(A, B, C, D) = \sum(1, 3, 5, 7, 13) + \Phi(2, 4, 6, 8, 12, 14)$$

15. Obtain the reduced expression for the function $f(x, y, z) = \sum(0, 2, 4, 6, 8)$
16. Obtain the minimal SOP expression for the function $f(A, B, C, D)$ whose truth table is given below:

Decimal value	A	B	C	D	f(A, B, C)
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	Φ
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	Φ
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	Φ
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	Φ

IMPLEMENTATION OF UNIVERSAL GATES

Integrated chips (IC) form an integral part of digital devices. The ICs are made up number of logic gates apart from other components. The usage of AND and OR gates cause difficulties and complicates the circuitry. Thus, the methods to reduce the number of logic gates that are employed in a circuit have to be evolved and considered. One such method is implementation of logic expressions using NAND and NOR gates. NAND and NOR can perform functions of all other logic gates and are hence termed as universal gates. We shall deal with implementation of NAND gates first followed by NOR gate in this chapter.

NAND IMPLEMENTATION

NAND gate is a universal building block. Since we have already dealt with the NAND gate elaborately, we shall now directly move on to the implementation of other logic operators using NAND.

Implementing an Inverter Using Only NAND Gate

There are two ways in which a NAND gate can be used as an inverter (NOT gate).

1. All NAND input pins connect to the input signal **A** gives an output **A**'. This is represented in Figure 6.1.

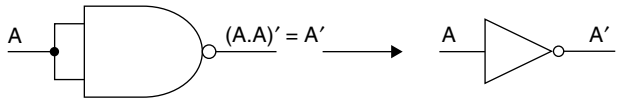


Fig.6.1 OR operator using NAND

2. In the second method, the input A is connected to terminal of the NAND gate and the all other terminal is given a HIGH logic (1), the output of the NAND gate will be inverted. This is represented in Figure 6.2.

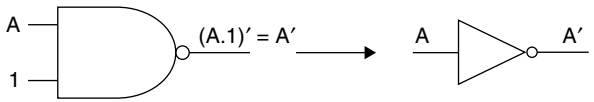


Fig.6.2 OR operator using 2 input NAND

Thus, a NAND gate can perform an operation of a NOT gate without using one.

Implementing AND Using Only NAND Gates

An AND gate can be replaced by NAND gates as shown in Figure 6.3. Consider an AND gate with both high inputs. Then the output of the AND gate is high. When such output is given to the NAND gate whose inputs are interconnected then it inverts it and the final output from a NAND gate is 0. Thus, an AND gate can be replaced by a NAND gate.

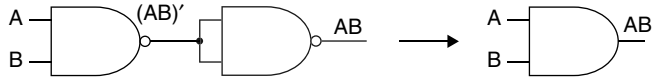


Fig.6.3 AND gate using NAND

Implementing OR Using Only NAND Gates

A NAND gate can reinstate an OR gate in a circuit. Consider two NAND gates, A and B the inputs of each are interconnected thus making the output an inverse of the input. Then the output from both the NAND gates is given to the third NAND gate whose inputs are also interconnected. The final output is equivalent to the output from that of an OR gate as shown in Figure 6.4.

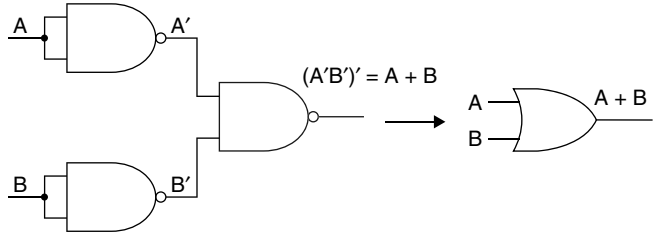


Fig.6.4 OR gate using NAND

Thus, the NAND gate is a universal gate since it can implement the AND, OR, and NOT functions.

NOR AS A UNIVERSAL GATE

A NOR gate besides NAND form the universal building block of many circuit as it also can substitute the AND, OR, and NOT logic operators. NOR gate executes any Boolean function without using the basic gates.

Implementing an Inverter Using Only NOR Gate

Figures 6.5 and 6.6 show two ways in which a NOR gate can be used as an inverter (NOT gate).

1. All NOR input pins connect to the input signal A giving an output A'.

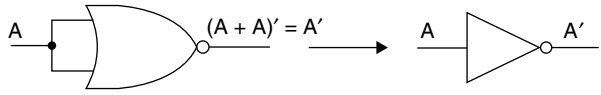


Fig.6.5 OR gate using NOR gate

2. One NOR input pin is connected to the input signal A, while all other input pins are connected to logic 0. The output will be A'.

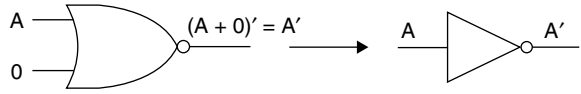


Fig.6.6 OR gate using 2 input NOR gate

Implementing OR Using Only NOR Gates

An NOR gate replaces an OR gate as shown in Figure 6.7. The OR is replaced by a NOR gate with its output complemented by a NOR gate inverter.

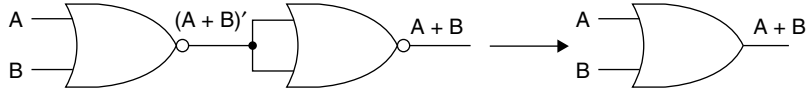


Fig.6.7 NOR replaces OR gate

Implementing AND Using Only NOR Gates

An AND gate is replaced by a NOR gate as shown in Figure 6.8. The AND gate is replaced by a NOR gate with all its inputs complemented by NOR gate inverters.

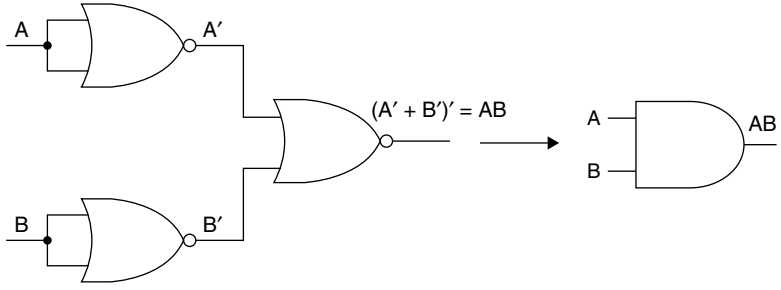


Fig.6.8 NOR replaces AND gate

Thus, the NOR gate is a universal gate since it can implement the AND, OR, and NOT functions.

EQUIVALENT GATES

There exist certain equivalent circuits to the basic operators like AND, OR, NOT. Figure 6.9 summarizes important cases of gate equivalence. The bubbles indicate a complement operation (inverter). The Boolean laws can be expressed using NAND or NOR as equivalent gates.

A NAND gate is equivalent to an inverted-input OR gate and is shown in Figure 6.9.

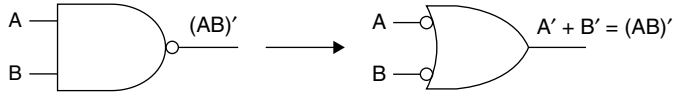


Fig.6.9 NAND gate is equivalent to an inverted-input OR gate

An AND gate is equivalent to an inverted-input NOR gate can be represented as in Figure 6.10.

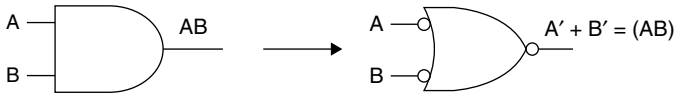


Fig.6.10 AND gate is equivalent to an inverted-input OR gate

A NOR gate is equivalent to an inverted-input AND gate and is depicted in Figure 6.11.

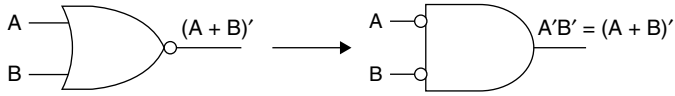


Fig.6.11 NOR gate equivalent to an inverted-input AND gate

An OR gate is equivalent to an inverted-input NAND gate. The logic symbol is shown in Figure 6.12.

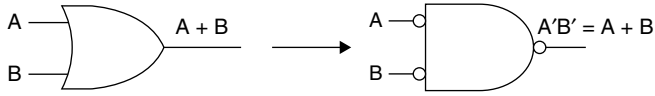


Fig.6.12 OR gate equivalent to an inverted-input AND gate

Two NOT gates in series are same as a buffer because they cancel each other as $A'' = A$. This is represented in Figure 6.13.

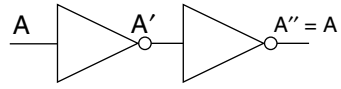


Fig.6.13 Double NOR gate equivalent to an inverted-input AND gate

NAND AND NOR IMPLEMENTATION TO BOOLEAN FUNCTIONS

Boolean functions can either be expressed in SOP or POS forms. We have already dealt with this extensively in the previous chapters. The same Boolean expression can be constructed using NAND and NOR gates in a two level implementation.

Consider a Boolean expression in the SOP form. In the SOP form, the AND gates will be in the first level, and a single OR gate will be in the second level. Boolean expressions in the POS forms, OR gates will be in the first level and a single AND gate will be in the second level. NAND gates are used to implement SOP forms and NOR gates are used to implement POS forms. This is best explained through examples.

Example 6.1 Implement the following SOP function

$$F = XY + Y'Z + XYZ$$

Solution: Being an SOP expression, it is implemented in 2-levels using AND-OR network as shown in Figure 6.14.

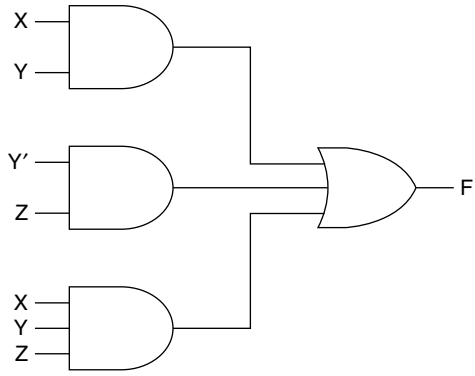


Fig.6.14 2-level implementation using AND-OR gates

To achieve the desired expression, two successive inversions, that is, OR operations are required but this might not be effective on a logic circuit. Therefore, one inversion is carried out at the output of AND gate and the second at the input of the OR gate. This leads to 2-level implementation and can be reducible by using NAND gates at both levels as shown in Figure 6.15.

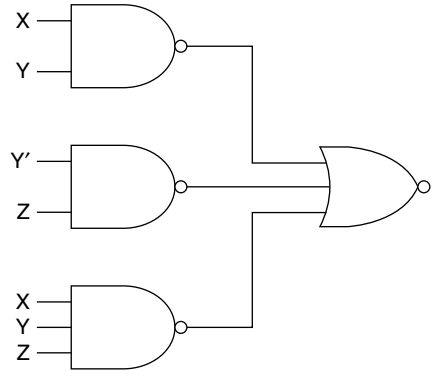


Fig.6.15 2-level implementation using NAND-NAND network

Example 6.2 Implement the following POS function

$$F = (X + Y)\,(Y + Z)\,(X' + Y' + Z')$$

Solution: The POS expression is implemented using OR gates followed by AND operation as shown in Figure 6.16.

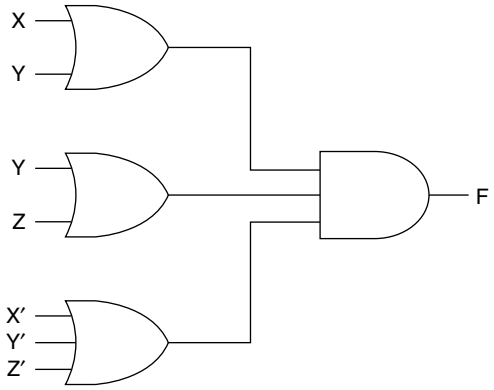


Fig.6.16 2-level implementation using OR-AND gates

To get the given POS expression, two successive inverters at the inputs of the AND gate must be performed. However, two successive inverters on the same line reduce the overall effect on the logic. Therefore, one inverter is used at the output of the first level OR gates and the other with the input of the AND gate. This results in a NOR operation and hence the AND and OR gates can be replaced by NOR gates as shown in Figure 6.17.

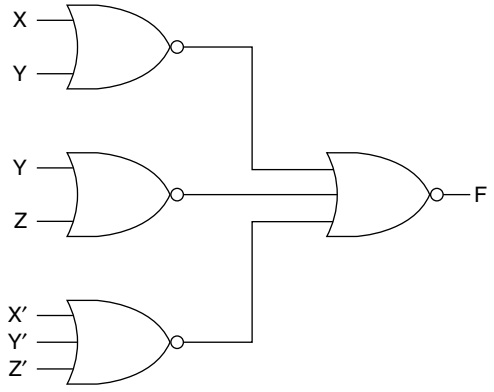


Fig.6.17 2-level implementation using NOR-NOR network

NAND–AND AND AND–NOR IMPLEMENTATION

Consider the Boolean expression F = XY + YZ + $\overline{XYZ}$, the given expression is in the SOP form and hence it has to be ANDed and then ORed. This can be achieved using a NAND- AND network Figure 6.18.

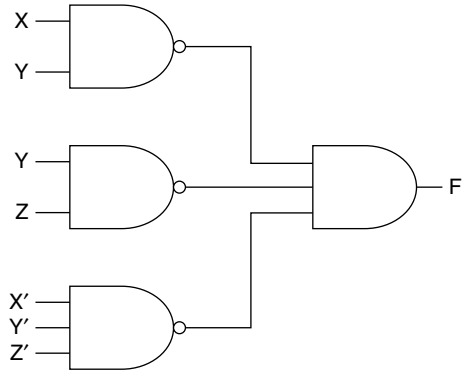


Fig.6.18 NAND–AND network

Since the AND-NOR is equivalent to NAND-AND circuit. The same logic expression can also be implemented using AND-NOR circuit as shown in Figure 6.19.

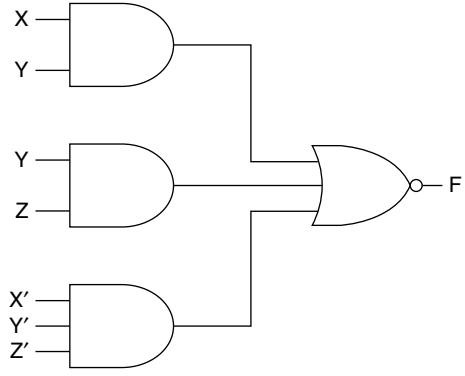


Fig.6.19 AND–NOR equivalence to NAND–AND circuit

NOR–OR IMPLEMENTATION AND OR–NAND IMPLEMENTATION

Consider a logic expression F = (X + Y) (Y + Z′) (X′ + Y′ + Z′), which is in the POS form. Since F is in POS form, it can be implemented by using NOR-NOR circuit. The complement of the output gives F, and this is achieved by using NOR-OR circuit as shown in Figure 6.20.

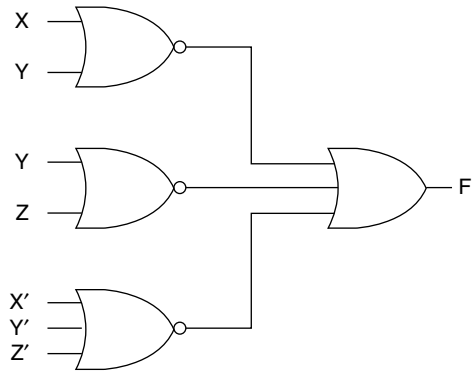


Fig.6.20 NOR-OR circuit

NOR-OR gate has OR-NAND as its equivalent circuit. Hence the same expression can also be implemented using OR-NAND circuit as shown in Figure 6.21.

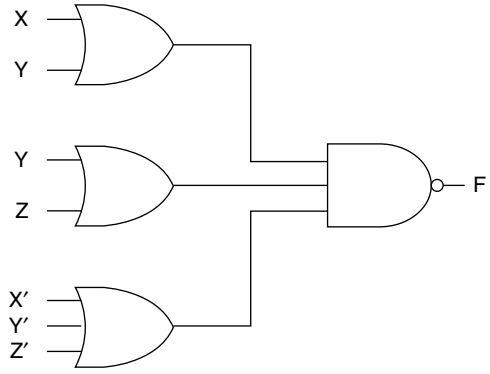
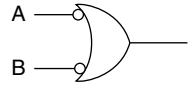


Fig.6.21 Equivalent circuit for NOR-OR

MULTIPLE CHOICE QUESTIONS

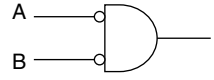
1. What is the output of the given logic symbol?



(a) $A\overline{B}$ (b) $\overline{A}B$

(c) $A + B$ (d) $A' + B'$

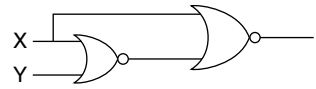
2. The output of the inverted AND gate is equivalent to



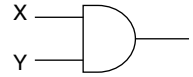
(a) A + B
(b) AB
(c) A′B′
(d) (A + B)′

3. When two NOT gates are connected in series and input A is given then output is
(a) A′
(b) A
(c) A + A
(d) 2A

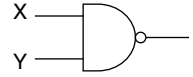
4. The logic circuit given below is minimised as



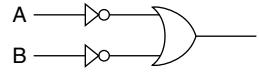
(a)

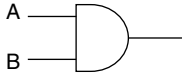
(b) 

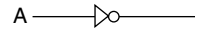
(c)

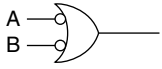
(d) 

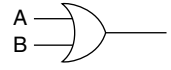
5. The given logic gate can be replaced using



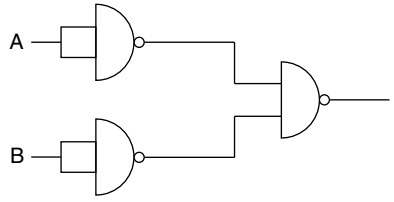
(a) 

(b) 

(c) 

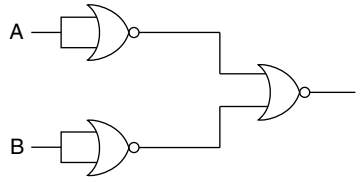
(d) 

6. The circuit given represents NAND implementation of _____ gate.



(a) OR
(b) AND
(c) EX-OR
(d) EX-NOR

7. The given circuit is the NOR equivalent of _____ gate.



(a) OR
(b) AND
(c) NAND
(d) EX-OR

Answers

1. (d) 2. (a) 3. (b) 4. (c) 5. (c) 6. (a) 7. (b)

DESCRIPTIVE QUESTIONS

1. Implement the expression $Y = ABC + AB\overline{C} + \overline{ABC}$ only using NAND gates.
2. Implement the expression $Y = \overline{A}B\overline{C}D + ABC + A\overline{D}C$ only using NAND gates.
3. Simplify the Boolean expression $Y = \overline{ABCD} + \overline{BC} + \overline{AD} + ABC$.
4. Simplify the Boolean expression and implement only using NOR gates
   (a) $X = C(B + C)(A + B + C)$
   (b) $Y = \overline{ABC} + \overline{A}B\overline{C} + A\overline{BC} + \overline{AB}C$
   (c) $\overline{XY + XYZ} + Y(X + \overline{Y})$
5. Simplify the Boolean expression and implement only using NAND gates
   (a) $(\overline{A} + C)(A + \overline{B})(B + C)$
   (b) $AB + \overline{A}B + AB\overline{C}(AC + AB)$

INTEGRATED CIRCUITS AND LOGIC FAMILIES

INTRODUCTION

Integrated circuits were invented by Jack Kilby in 1958 forming the genesis of almost every electronic product used today. From cell phones, to modems, to Internet audio players and many more, the chip has revolutionized the electronic world and enabled an entire industry to grow. An integrated circuit or monolithic integrated circuit also abbreviated as IC. A chip, or microchip is a collection of various discrete components such as passive as well as active elements fabricated on a single semiconductor crystal chip. The semiconductor material used for fabricating IC chips is normally silicon. The IC chips can be constructed much smaller than the discrete circuits that is made from independent components. The first invented IC comprised of only a transistor and other components on a slice of germanium. They can be made as compact as to encompass several billion transistors and other electronic components in a small area.

The main advantages of ICs over discrete circuits are as follows:

- **Performance:** The components in an IC chip switch faster between the logic states. Thus, this consumes less power. This could also be attributed to the components in close proximity. For example, a typical chip areas range from a few square millimeters to around 450 mm^2, with up to 9 million transistors per mm^2.
- **Cost:** Since the IC chips are constructed on much less material as compared to discrete circuits, the cost is effective. Discrete circuits use individual resistors, capacitors, diodes, transistors and other devices to achieve the circuit function. These individual or discrete parts must be interconnected on a circuit board. The board, assembly, soldering and testing adds to a part of the cost. This method, however, increases the cost of the circuit. On the other hand, although the circuit boards are not eliminated in ICs, the number of components are reduced and also combined as a single unit using photolithography principle thereby reducing the cost.

CLASSIFICATION OF ICs

The integrated chips are classified under the following categories:

- Based on the circuit size, they are divided into SSI, MSI, LSI, VLSI, ULSI, etc.
- Based on the design style, they are divided into standard cell, gate array, etc.
- Based on the logic circuit technology, they are classified into TTL, CMOS, etc.
- Based on the design type, they are classified into analog, digital and mixed.

CLASSIFICATION BASED ON SIZE

The number of components on the IC chip enables us to classify them into the following sub classes. This is done on the basis of Moore's law that states the number of components that can be packed on a computer chip doubles every 18 months while price stays the same.

- **Small Scale Integration (SSI):** This method of integrating components is adopted when the number of transistors is about 1–100 on a single chip. E.g., Gates, flip-flops, operational amplifiers.
- **Medium Scale Integration (MSI):** This technology was developed for integrating around 100–1000 transistors on a single chip. E.g., Adders, counters, MUX, 4-bit microprocessors, etc.
- **Large Scale Integration (LSI):** The technology was developed by integrating 1000–10,000 transistors on a single chip. E.g., 8-bit microprocessors, ROM, RAM.
- **Very Large Scale Integration (VLSI):** The technology was developed by integrating 10,000–1 million transistors on a single chip. VLSI is the process of creating integrated circuits by combining thousands of transistors into a single chip. VLSI was introduced in the 1970s to overcome the complexicity of semiconductors and communication technologies. The 16–32 bit microprocessor, CMOS, etc. are examples for VLSI.
- **Ultra Large Scale Integration (ULSI):** The technology was developed for integrating 1 million–10 million transistors on a single chip. E.g., Special purpose processors.
- **Giant Scale Integration (GSI):** The technology was developed by integrating the number of transistors of above 10 millions on a single chip. E.g., Embedded system, system on chip.

COMMON DIGITAL IC TERMINOLOGY

So far, logic gates have been mentioned in the logic HIGH or LOW levels. However, in the logic technologies, it is essential that the voltage and current levels are mentioned. Some of the commonly used terminologies in IC logic technologies are as follows:

- **V_{IH}:** It is the input voltage at HIGH state that corresponds to logic 1 at input.
- **V_{IL}:** It is the input voltage at LOW state that corresponds to logic 0 at input.

- **V_{OH}:** It is the output voltage at HIGH state that corresponds to logic 1 at input.
- **V_{OL}:** It is the output voltage at LOW state that corresponds to logic 0 at input.
- **I_{IH}:** It is the input current flowing from the input when the corresponding input voltage is at HIGH state or logic 1.
- **I_{IL}:** It is the input current flowing from the input when the corresponding input voltage is at LOW state or logic 0.
- **I_{OH}:** It is the output current flowing from the output when the corresponding output voltage is at HIGH state or logic 1.
- **I_{OL}:** It is the output current flowing from the output when the corresponding output voltage is at LOW state or logic 0.

CHARACTERISTICS OF LOGIC FAMILIES

Fan-In

The number of inputs to a gate is referred as Fan-in. Consider a 3-input OR gate. It has a fan-in of 3. A NOT gate always has a fan-in of 1 since it has only 1 input. For a given logic circuit fan-in is the number of inputs given to the circuit. The delay produced by a gate can be attributed to the fan-in. This is due to the fact that the delay produced by the logic circuit increases as a quadratic function of fan-in. Thus, the number of inputs that enable the logic operation of the circuit is defined as fan-in. Most inputs have a fan-in of 1.

Fan-Out

The fan-out of a gate refers to the load that is connected to the output which does not change the operation of the gate. Load here refers to the magnitude of current that is required to operate the input of another gate in the same logic family. Fan out is sometimes also referred as loading and is completely restricted within one logic family. When there are two or more logic circuits that are interconnected, so the fan-out could be capable of driving both the logic families. Fan-out is an important characteristic since the amount of current supplied by the gate is limited for each operation. In other words, if the logic gate has a fan-out of 3, then it can drive 3 logic inputs without perturbing the normal functioning of the gate. Thus, fan-out is defined as the number of output that drives the input of the gate without disturbing the logic HIGH or LOW states.

Noise Immunity and Noise Margin

Undesirable voltage signals from surrounding networks sometimes pass through the wires used for connecting logic circuits. These unwanted stray signals from electric sources are defined as noise. Noises can be from DC or AC source. DC noise is from drift voltage and AC noise is generated while switching signals. Noise margin is the threshold level that a gate can endure without causing any change in its output and beyond which it can damage the logic circuit. It is also referred as safety margin and is highly important as it exhibits the ability of the device to function properly in a noisy

environment. The voltage levels of input signal and output signal help to calculate the noise margin. Noise margin is calculated for both low and high voltage levels. The low noise margin is defined as the difference in the maximum amplitudes of the input and output voltages at LOW intervals.

$$LNM = V_{IL} - V_{OL}$$

The high noise margin is defined as the difference in the maximum amplitudes of the output and input voltages at HIGH intervals.

$$HNM = V_{OH} - V_{IH}$$

Thus, noise margin of a digital gate is then defined as:

$$NM = \min \{HNM, LNM\}$$

Noise immunity is defined as the amount of noise level a logic circuit can withstand without being amplified more than unity gain.

Time Delay

A logic system comprises of number of logic gates cascaded together. The transition from one gate to another causes delay in the transition time. The delay depends on the number of logic gates in the circuit. The speed of the operation is important for any circuit. The speed can be improved by reducing the logic components in the circuit. Increase in the speed of operation results in reduced signal delay and faster transmission. Many parameters contribute to the time delay in a logic circuit.

- **Logic gate delay:** In the logic system, the signal has to pass through the logic gates in the circuit before it finally reaches the output terminal. During this process, the gate may introduce a delay in propagation and hence also known as the propagation delay. All transistors do not have the same propagation delay and hence, the delay is calculated at the midpoints. The gate delay may be considered in two phases. The delay in the time from the turning on of the circuit where the transmission is from low to high is called turn-on delay and the delay during the transmission from high to low is called the turn-off delay. For a single transistor, the propagation delay is the sum of turn-on delay and turn-off delay. Since the circuit consists of many transistors and gates cascaded together, the propagation delay will be a cumulative of delays. Hence, the propagation contributes to a major portion of the delay and it is of major concern. An excessive delay sometimes hinders the operation of gate. The propagation delay is generally increased when logic operations are performed at low voltages.
- **Wire delay:** The signals in the circuit are transported from one point to other through wires. These wires bring in a delay and it increases with the frequency specifically for sub micron sized transistors. This delay is also knows as transport delay or flight time.
- **Rise and fall time:** This is the time required by the gate to change its output from 10 to 90 % or 90 to 10 % voltage levels. These values are often mentioned by the manufacturers of logic gates and other digital components as the suitable frequency can be calculated and the gates can be used in that frequency.

Power Dissipation

Power is required to operate any electronic circuit. Power applied to the main supply is different from the power delivered to the gate. The power supply voltage is always denoted as V_{CC} in case of TTL devices, and for MOS devices it is denoted as V_{DD}. Let us consider the power supply to be V_{CC} and the current that passes from the source to the gate as I_{CC}, then the dissipated power is given as $V_{CC}I_{CC}$. For an individual gate, the power dissipation varies from 0.05 mW to 50 mW. The current from the source while passing through the gate causes the transition current. The transition current (I_{CCT}) is taken out of the logic gate during HIGH to LOW transition or transmissions from LOW to HIGH. The transition current for TTL is more when compared to gate current at LOW and gate current at HIGH. However, the same transition current is less compared to the current passing through gate at LOW and HIGH. Power can be of two components: static power dissipation and dynamic power dissipation.

1. **Static power dissipation (P_s):** This is caused by leakage current and this is consumed when the gate is HIGH or LOW. The leakage current contributes to a major amount of the total power dissipation.
2. **Dynamic power dissipation (P_d):** The power dissipated during input and output transitions is referred as P_d dynamic power dissipation. Since the dynamic power dissipation includes the power due to leakage and power during transitions, it can also be considered as the total power dissipation. Total dissipated power is given as:

$$P = P_s + P_d$$

The total power dissipation is the sum of static power dissipation and dynamic power dissipation.

Power dissipation should be controlled as it defines the battery life. If the power dissipation keeps increasing then it would reduce the life of the battery. Sometimes power is dissipated as heat generated by the chip. Excessive heat dissipation can cause error in the output values as the normal operation of the gates can be disturbed.

SEMICONDUCTOR DIODES AND TRANSISTORS

Electronic switches are an essential of any circuits. The most commonly used switches are semiconductor diodes and transistors. Before dealing with the switching action of diodes and transistors we shall have a brief introduction on semiconductors, the non-linear models of diodes and transistors, their load line analysis and application. We will first deal with diodes followed by transistors.

Materials are classified into conductors, semiconductors and insulators depending on their electrical conductivity. Semiconductors are materials whose electrical conductivity lies between conductors and insulators. They have introduced tremendous importance in technologies and are also economical. Silicon is the most commonly used semiconductor material to create commercially available electronic circuitry. The components made up of these semiconductor materials find their application in electronic devices from computer systems to digital devices. Figure 7.1 depicts the lattice arrangement for silicon (Si).

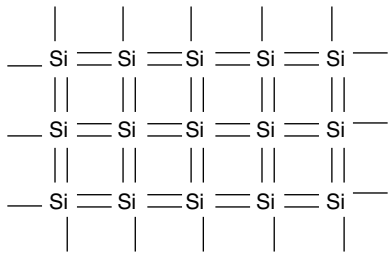


Fig.7.1 Lattice arrangement for silicon

Semiconductors materials are of two types: intrinsic semiconductor and extrinsic semiconductor. The purest form of semiconductor is known as the intrinsic semiconductor. When suitable energy is applied to the atoms in the crystal lattice, the atoms vibrate and some of the valence electrons move to the conduction band. The free flow of electrons results in a current flow in semiconductor. The conductivity of intrinsic semiconductors can be improved by adding impurities. The process of adding impurities to the crystalline structure of the semiconductor is known as doping. The same intrinsic semiconductor can be converted to extrinsic semiconductor by adding the impurities. Depending on the type of impurity added, they are further classified into N-type and P-type semiconductors.

When a pentavalent impurity like phosphorous is added to the semiconductor, each of the phosphorus atoms bond to one silicon atom in the outermost orbit. There exists an extra electron of phosphorus that is in the lattice and is called donor. This is because phosphorous has five valence electrons. Accumulation of these extra electrons results is increase in negative charge of the material and hence, they are termed as N-type semiconductors as shown in Figure 7.2.

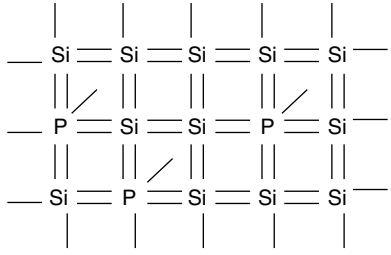


Fig.7.2 Silicon doped with phosphorous

When a trivalent impurity like boron is added to the semiconductor a P-type semiconductor is formed. This is due to the fact that boron has three valence atoms that form covalent bonds with the four silicon atom in its outermost orbit. This is shown in Figure 7.3. Vacant or missing places in the lattice is called hole. Hole is similar to an electron except that it carries a positive charge and hence is acceptors. Silicon doped with acceptors is also called a P-type semiconductor because of the abundance of holes.

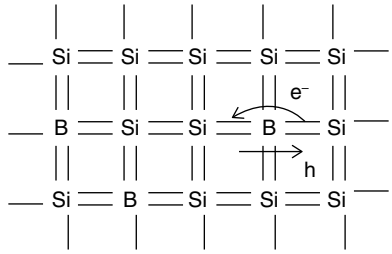


Fig.7.3 Silicon doped with boron and formation of hole

Since the electrons and holes move across the lattice they are referred as charge carriers and their movement is called mobility. In the N-type semiconductors, electrons move freely and hence, they are the major charge carriers. For a P-type semiconductor, the holes are the majority charge carriers. When the P-type and N-type semiconductors are fused together they result in electronic devices. The contact between a P-type and N-type material results in the PN junction.

The region where the P-type and N-type come in contact is called the depletion layer and is a non-conducting layer. This is because the electrons from the N-type and holes from the P-type attract each other and move towards each other. They eliminate each other and hence, it is called the depletion layer. The process of eliminating the charge carriers is called recombination. The combination of N-type and P-type semiconductors results is a junction diode.

SEMICONDUCTOR DIODES

Diodes are electrical devices that conduct electricity only in one direction. The arrow represents the positive terminal and the bar represents the negative terminal. It can act as an electrical switch as it conducts only in one direction, that is, it conducts only in the forward biased condition. Biasing refers to the voltage applied to the junction. Since the diode allows current only in one direction it is similar to an electronic valve. Diodes are used for the conversion of AC to DC. It is used to protect circuits as they limit the voltage flowing into the circuit. Silicon and germanium are the most commonly used semiconductor materials to make diodes. Diodes are of many types; we will restrict our study to PN junction diodes.

PN Diodes

These are the normal diodes and are made of silicon sometimes using germanium. Earlier cuprous oxide and later selenium was used to make these diodes. The PN diode or junction diode conducts current only in one direction and hence, can be used as a switch. The symbol and actual image of a diode is shown in Figure 7.4. The semiconductor diode consists of an anode and cathode. The anode is a P-type semiconductor and cathode is a N-type semiconductor.

Fig.7.4 The PN junction

The diode can function as an electronic switch. When a positive voltage is applied to the diode it is forward biased and conducts.

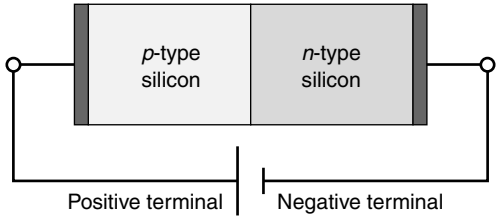


Fig.7.5 Forward biasing a diode

The diode ideally has zero resistance. However, when a negative voltage is applied, then the diode shows high resistance. This is similar to the characteristics of an electronic switch. When the switch is closed it shows zero resistance and when it is open it shows infinite resistance. Although the diode has similar characteristics as the switch, it is not considered as an ideal device.

Working of PN Diode

When the diode is given a negative bias, that is, the positive terminal of the diode is given the negative voltage and negative terminal of the diode is applied positive voltage, the diode does not conduct and is hence in OFF state.

The voltage-current characteristic of diode is shown in Figure 7.6. It is evident from the figure that the curve is not a straight line. Instead of following Ohm's law, the PN junction diode follows an exponential law and hence, is non-linear in

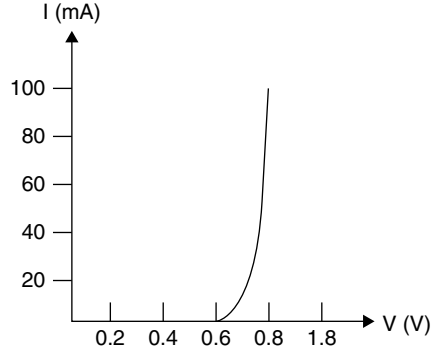


Fig.7.6 Voltage-current characteristics of diode

behaviour. The diode in the forward bias should have a voltage of 0 V for 3 milliamperes since R = V/I = 0/3 = 0 Ω but the diode shows approximately 0.75 V for the same current which is seen in the graph. Therefore, the actual resistance is R = V/I = 0.75/3 = 250 Ω. However, this resistance is not too large and diode can still be used as a switch.

The second problem that the diode encounters while functioning as a switch is the voltage drop. It has a voltage drop of 1.4–1.7 V for each cell. The high forward voltage drop results in a lower efficiency. The voltage drop for silicon diode is 0.7 V. Multiple cells are stacked to increase the peak voltage. Diodes take time to switch between the ON and OFF states. The switching speed is generally negligible. This problem is overcome by using Schottky diodes. These are metal semiconductors diodes. The storage time of Schottky diodes is very less and the recovery time is in picoseconds.

The response time of the diode is affected by two types of capacitance. The diode in the reverse biased condition is similar to a capacitor with the two terminals acting as the conductor and the junction or depletion layer acting as the insulator. When the voltage keeps increasing, it increases the barrier width and hence, the capacitance decreases. This can be solved by using varactor diodes that have varying capacitance of 3–100 picofarads (pF).

Apart from the junction capacitance, the diode exhibits another type of capacitance that arises due to diffusion. Assume the P-type material to be highly doped and the N-type material less doped. When a positive biasing is applied, the holes from P-type recombine with the electrons in the N-type material but as the number of electrons is less compared to the holes, the holes get accumulated in the depletion layer. This is similar to charging of the capacitor.

Similarly for a negative bias, the electrons move towards the holes in the P-type material and recombine. This causes a current flow that flows even after the biasing is off. This is similar to discharging of a capacitor. This causes diffusion capacitance. The above two capacitances restrict the diode from being an ideal switch. These diodes are vastly used in CMOS integrated circuits with two diodes used per pin and the other being internal diodes.

SEMICONDUCTOR TRANSISTORS

Transistor is a semiconductor device made of either silicon or germanium. It has three terminals unlike the diode that has only two terminals. It is of two types: NPN or PNP transistors. A transistor performs two functions: it amplifies the input signal and can be used as an electronic switch. The three terminals are emitter, base and collector. The voltage or current applied to one terminal of the transistor is obtained amplified at the other terminal. This is because the power of the output is much more than the power of the input and thus the transistor amplifies the signal. The transistor overcomes the problems faced by the diode as a switch. By applying a small control current or voltage to the third terminal of the voltage or current across the other two terminals can be controlled. Such a device is a transistor switch. Integrated circuits have many transistors embedded in them and are made of silicon but single

individual transistors are also available. Transistors can allow small current that can control larger current. Such transistors are known as current regulators. The amount of current flowing across base and emitter determines the amount of current between collector and emitter.

Transistors are mainly of two types: bipolar transistors and field effect transistors. The bipolar junction transistors will be dealt more since they form the basis of interface elements to microcontrollers.

Bipolar Transistor (BJT)

Bipolar transistor (BJT) is a three terminal current controlled device. It consists of either two N-type or one P-type layer of the semiconductor material forming the NPN transistor or two P-type and one N-type layer of the semiconductor material forming the PNP transistor. The base is narrower than the emitter. Since the current has to flow across both *p* and *n* terminals it is called bipolar transistors. Three sections of semiconductor material having different doping concentration are used to form a BJT. The construction, symbol and nomenclature of the two types of BJT are shown in Figure 7.7.

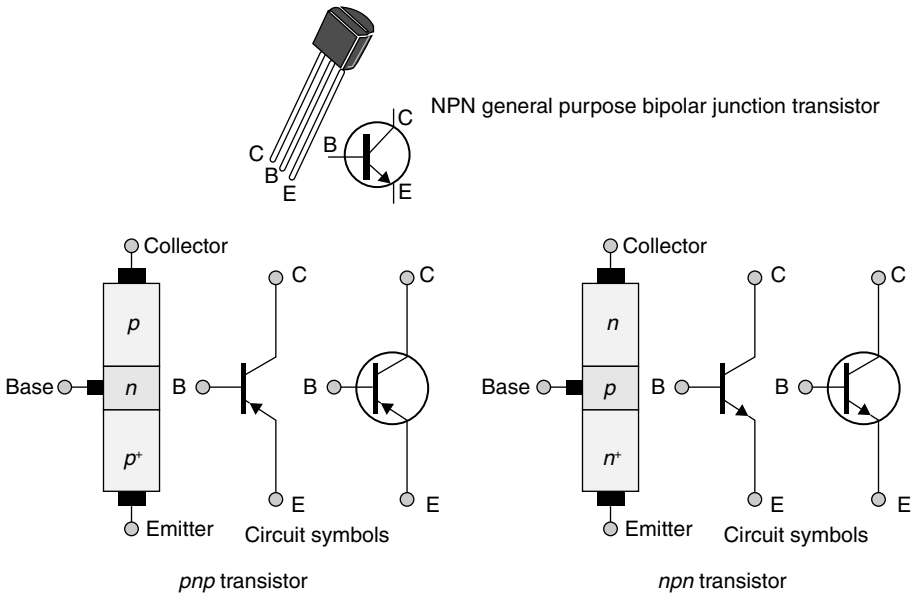


Fig.7.7 Symbol and nomenclature of BJT

Let us consider the working of NPN BJT transistor. This is formed by fusing two PN junctions back to back. The base-emitter (BE) acts as a diode in the forward biasing condition. The holes from the base flow to the emitter with the collector open. Few holes and electrons recombine and the remaining charge carriers contribute to net flow of base emitter current. Now if the base collector junction is reverse biased, electrons from the emitter region reach the base and recombine. The remaining electrons are collected by the collector. Owing to the fact that the base is narrow, the electrons gain

momentum and cross to the collector and this results in a net flow of current from collector to emitter in addition to the hole current from base to emitter. The collector-base current is larger than base-emitter current. An important property of the BJT is that the base is applied with small current that controls the transistor's collector current. The voltage-current flow in BJT is represented in Figure 7.8.

$$I_E = I_B + I_C$$
$$I_C = \beta I_B$$

β is current amplification factor. It depends on the physical properties of the BJT. The current amplification factor β ranges from 20 to 200.

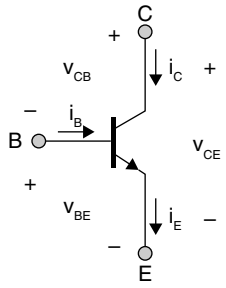


Fig.7.8 Voltage-Current flow in BJT

The operation of the device can be explained using two voltages and two currents. BJT has a family of curves instead of a single curve. For each voltage current source, an individual I–V characteristic is to be studied. One characteristic is for base current to base emitter voltage while the other for collector current and collector–emitter voltage. Figure 7.9 explains the V–I characteristic of base current to BE voltage when the collector is open and is similar to diode. The base current IB induces forward bias. The BE voltage is obtained by varying base current.

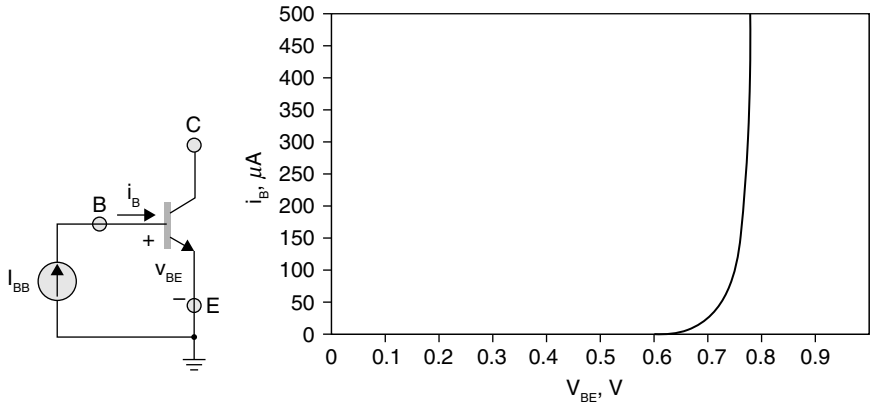


Fig.7.9 V–I characteristic of base current to BE voltage

When the voltage is applied across the collector emitter junction then the collector-emitter voltage and collector current can be varied along with the base current. The circuit and corresponding V–I characteristics is depicted in Figure 7.10.

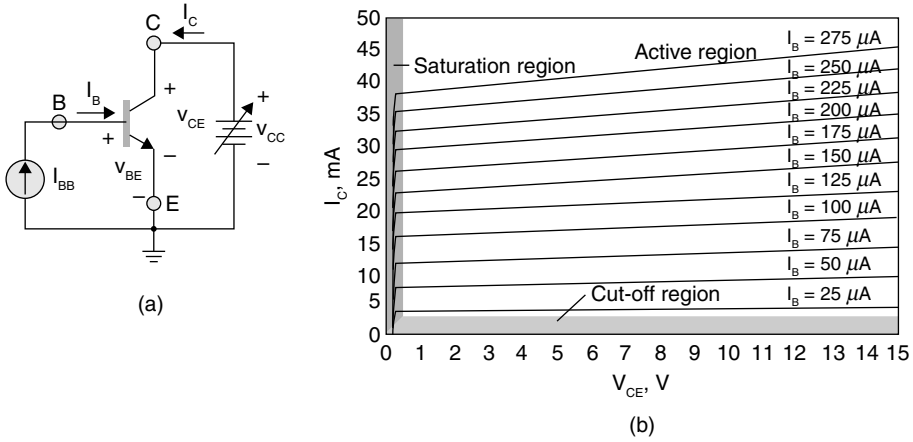


Fig.7.10 V–I characteristics of collector–emitter voltage and collector current

The curve shows the following three operating regions:

1. **The cut-off region:** This region occurs when both the junctions are reverse-biased. The base current is very small and essentially no collector current flows. Thus, the transistor is OFF in this region.
2. **The saturation region:** This region occurs when both junctions are forward biased.
3. **The active linear region:** The BE junction is forward-biased and the CB junction is reverse-biased thus making the transistor to act as a linear amplifier.

Field-Effect Transistor (FET)

The field-effect transistor (FET) has three terminals, namely, source, drain and gate. It is a unipolar device as it can have any channel which has either electron (N-channel) or hole (P-channel) conduction. FETs are more temperature stable and smaller than BJTs therefore making them particularly useful in integrated-circuit(IC) chips. The symbol of FET is shown in Figure 7.11.

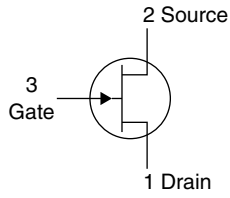


Fig.7.11 Symbol of FET

The voltage is applied across source and drain and current flows through N type material. When the voltage is applied across gates G_1 and G_2, a depletion region is formed around the gates. This results in restricting the N-channel current flow. The larger the gate voltage, the narrower the channel and thereby can cut-off the channel. When both the gates are connected together and voltage is applied to the gate, the same effect is exhibited. This controls the drain current. It is also known as the junction field effect transistor (JFET). Their impedance is very high around 1 MΩ.

Metal Oxide Semiconductors

Transistors are generally assembled on semiconductor substrate of silicon. The pure silicon semiconductors have poor conductivity since they have no free carriers. To improve their conductivity, impurities are added. We have dealt with this process elaborately above. Metal oxide semiconductors (MOS) are created by superimposing several layers of conducting, insulating and transistor-forming materials. It has three terminals: source, drain and gate. An N-type material having relatively lower resistivity is diffused into a P-type material. The P-type material forms the substrate. The silicon substrate is converted into an insulator by oxidizing it into silicon dioxide. Source and drain are incised into the substrate through the insulator forming the electrodes. The electrodes are used for external connections. Gate is separated from the substrate by the insulator. There are two types of MOS transistors:

1. ***n*-MOS:** Negatively doped silicon, rich in electrons and
2. ***p*-MOS:** Positively doped silicon, rich in holes.

The symbol of *p*-Mos and *n*-Mos transistors is shown in Figures 7.12 and 7.13.

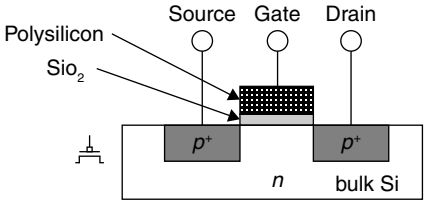


Fig.7.12 Symbol of *p*-MOS transistors

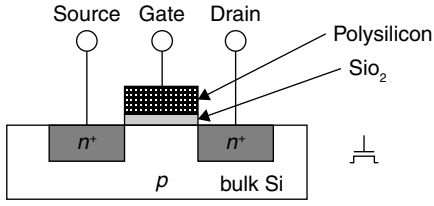


Fig.7.13 Symbol of *n*-MOS transistors

When a positive voltage is applied to the drain, then current flows across the source and drain terminals which is similar to the current flow across collector to emitter of a open bipolar junction transistor.

When a positive voltage is applied to the gate, it attracts minority electrons to accumulate beneath the gate. This will increase the source drain current since the carriers are increased. Thus, the current flow is enhanced and is called enhanced mode of MOSFET.

A slight variation in the fabrication of MOSFET can result in the flow of current even when the gate voltage is 0V. A thin slice of N-type material is infused into the substrate. The source drain current can be decreased by applying negative voltage to

gate. This causes the electrons to flow into the P-type substrate that result in depletion. This is known as the depletion mode of MOSFET.

MOS transistors can be treated as simple ON-OFF switches. The source (S) and gate (G) terminals control the operation of the switch. Consider the high voltage to be represented by 1 and low voltage is given as 0.

Consider an *n*-MOS transistor. When the gate is given logic low (0), then the drain and source act as an open switch and does not conduct. Thus, no current flows. When the gate is given logic high (1), then the source drain channel is closed and the switch conducts and the current flows.

Consider a *p*-MOS transistor. When the gate is maintained at 0, then the switch is closed and conducts, on the other hand, when the gate is at logic high (1) then the switch is open and does not conduct. The switching operation of the MOS transistors is shown in Figure 7.14. Thus, the working of a *p*-MOS transistor is the reverse of an *n*-MOS transistor.

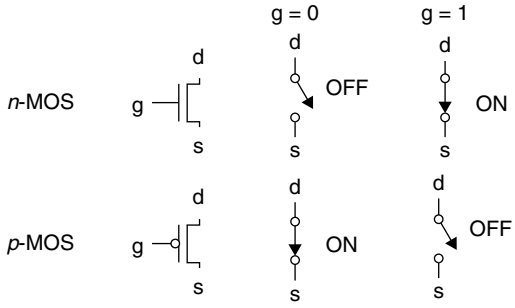


Fig.7.14 Switching operation of *n*-MOS and *p*-MOS transistors

MOSFETs have excellent low power requirements less than 5 microwatt and the input impedance of MOSFET is exceedingly high of the order of 10^{12} Ω making it an ideal logic device. But the high input impedance has a drawback on the device. It causes the time constant of capacitors thereby increasing the response time.

LOGIC FAMILIES

A logic family is a set of devices and techniques used to implement logic. There are various logic families available. Number of enhancement has been developed in the operation of these logic families. The major logic families, their characteristics, advantages and disadvantages are discussed.

Diode Logic

The diode logic uses diodes and resistors to implement Boolean functions. Diodes function as switch that conducts only during forward biasing condition. The simplicity in the circuits is the main advantage of diode logic but it lacks amplification of the input signal. This limits the application of diode logic. The diode logic can implement only logic OR and logic AND operations that are non-inverting. All other

logic operations cannot be implemented using diode logic. The diodes and resistors are either connected in parallel or series that function as electronic switches controlled by input logical variables. When the diode is forward biased the switch is closed and when the diode is reverse biased the switch is open.

DL OR Gate: The diode OR gate has two inputs and an output. The output is high only if at least one input is high and is low when both the inputs are low. The circuit for a basic DL OR is shown in Figure 7.15. The OR logic gate is implemented in the diode logic by connecting the diodes parallel. The diodes are normally open switches. It is clear from the circuit that the input is applied to the anode of the diode and the output is taken at the cathode of the diode across the resistor R_1 to ground. The +5 volts always represent logic 1 or ON, and ground, or zero volts a logic 0 is represented as in Figure 7.15.

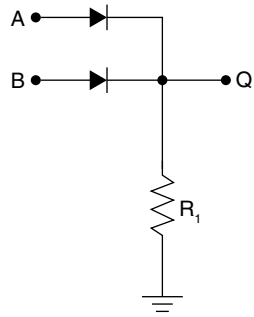


Fig.7.15 Diode logic OR gate

- **Inputs at logic 1:** In the circuit, if both the inputs are connected to +5 V or logic high input logical 1 then the diodes are forward biased and are closed. The current passes across the diode and causes high voltage drop across the resistor R_1. The output is at high or logic 1. If one input is high and the other input is low, then the diode connected to the high input acts as closed switch and the output is still high.
- **Inputs at logic 0:** When both the inputs are connected to zero volts or ground, then the inputs are at logic 0. Both the diodes are reverse biased and switches are open and do not conduct. Therefore, the output across the resistor will also be zero volts.

Thus, the diode logic implements the OR functions. Assume that the diode induces error in the circuit. The silicon diode introduces a voltage drop of 0.65 V–0.7 V therefore even when the voltage is not exactly +5 V but +4.35 V at the output still the diodes are said to be conducting. We consider the noise margin in this case. Thus, any voltage above +3.5 volts is considered to be logic 1, any voltage below +1.5 volts shall be logic 0. Output voltages between +1.5 and +3.5 volts fall under the undefined voltage region.

DL AND Gate: The DL-AND gate is implemented by connecting open switches in series. Since diodes are two terminal devices, they cannot be driven by grounded input voltage sources when connected in series. Thus to overcome this problem, the diodes are connected in parallel as in the case of OR diode gates with modification. To obtain AND logic instead of OR function the input and output logical variables are inverted

or reversed. The circuit of DL AND gate is shown in Figure 7.16. In a diode AND gate, if both the input voltages are high then the output voltage is high. If both the inputs are low and one is low then the output voltage is low.

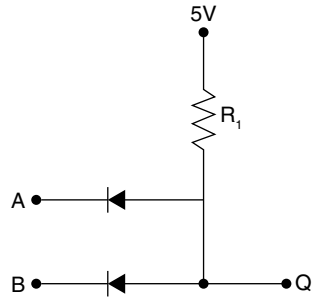


Fig.7.16 Diode logic AND gate

Consider the circuit for AND logic gate, the input to the diodes is connected in opposite direction. The input voltage flows across the resistor thus creating the current to flow across the anode of the diodes and then to its cathode. The output is taken across the resistor R and ground terminal. This gives the complementary voltage drop.

- **Inputs at logic 1:** When both the inputs A and B are at high, then the diodes are neutralized. There is no voltage drop across the diodes and the diode switches are open. Since no current flows across the resistor and no voltage drop, the output is high. Thus, the operation of diode logic AND is the reverse of diode logic OR gate since the diodes are reversed.
- **Inputs at logic 0:** When both the inputs are at low, the biasing voltage (+5 V) flows through the resistor and reaches the diodes and finally the input source. This causes the diodes to be forward biased and the diode switch is closed. Thus, the output voltage drop across the diode is logic 0. If one of the inputs is high and the other is maintained at low, then the diode connected to high input voltages or logic 1 is reverse biased and its input voltage is disconnected from the output. The output is again 0. Thus, this circuit performs the logical AND functions.

Disadvantages of Diode Logic

- Diode logic only permits the OR and AND functions.
- Diode logic suffers from voltage drop from one stage to the next. Unlike the active logic gates that can be integrated to larger components, the diode logic cannot be cascaded as they face few problems.
  - The diodes possess a voltage drop of 0.65 V during the forward bias condition. When the identical diodes are cascaded, this voltage is either added to or subtracted from the input of every gate. Thus, this voltage accumulates at each stage. For an OR gate, the voltage drop decreases the high voltage level (logic 1) while in an AND gate, it increases the low voltage level (logic 0).

  - The second problem the diodes face during cascading is that of the voltage divider. The input voltage sources possess an internal resistance that joins the gate resistor to produce a voltage divider action on the voltage levels. The source resistance decreases logic 1 level in an OR gate. In an AND gate, the gate resistance increases logic 0 level.
- Diode logic is used extensively to produce simple logic functions but not in integrated circuits.
- These issues restrict the usage of diode logic and have made them obsolete.

Resistor–Transistor Logic (RTL)

The resistor–transistor logic (RTL) are digital circuits that are constructed using resistors and bipolar junction transistors (BJTs). The resistors are the input network while the transistors function as switching devices. The disadvantage of DL is overcome using RTL since the transistors not only operate as switch but also amplify signals. RTL family was the first developed transistor logic circuits that were later improved to form the other classes like diode–transistor logic (DTL) and transistor–transistor logic (TTL). The introduction of RTL family revolutionized the circuit technology by constructing the first monolithic integrated circuit.

A RTL inverter or NOT gate is shown in Figure 7.17.

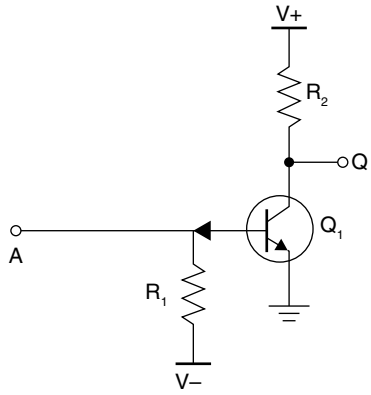


Fig.7.17 RTL NOT gate

Bipolar transistor switch is the simplest RTL gate. The resistor R_1 in the circuit is used across the base and input terminals. This resistor increases the voltage drop from 0.7 V to 1 V by converting the input voltage into current. The resistance R_1 is chosen in such a way that it saturates the transistor and obtains high input resistance. The collector resistor R_2 converts collector current into voltage. The resistance of R_2 is high to saturate the transistor and low to obtain output resistance.

The RTL NOT gate can be converted into a RTL NOR gate by introducing additional base resistors R_3 and R_4. The circuit of an RTL NOR gate is shown in Figure 7.18.

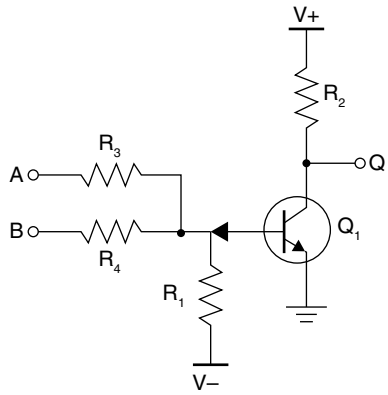


Fig.7.18 RTL NOR

When both A and B are given logic 0, then the transistor is cut-off. The output is inverted since it is complementary. This is because the voltage drop across the collector–emitter junction of the transistor Q_1 is taken at Q instead of taking it across collector resistor R_2. A multistage NOR gate circuit is shown in Figure 7.19. It consists of parallel connections of BJTs that are controlled by input logics. The inputs are not interconnected and hence if A is high, transistor Q_1 conducts and pulls Q to ground. Similarly, if B is high then transistor Q_2 conducts and pulls Q to ground.

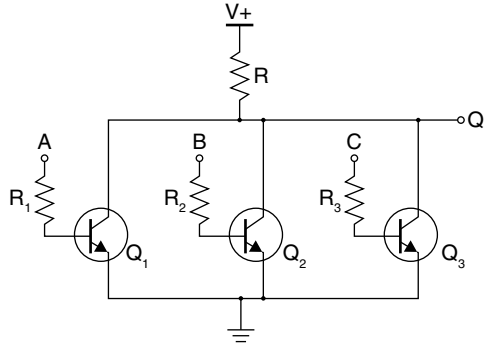


Fig.7.19 Multistage RTL NOR gate

Advantages and Limitations: Minimum transistors are required to implement logic expressions. RTL leads to the much improved logic families like diode–transistor logic and then transistor–transistor logic. The disadvantage is that the RTL has high power dissipation especially when the transistor is at logic 1.

Diode–Transistor Logic (DTL)

DTL logic family uses both transistors and diodes. A DTL NAND gate is shown in Figure 7.20. Each input to the circuit is let through a diode. The AND gate is formed

by the diodes and 5K resistors. Since the logic family has transistor that acts as a current amplifier.

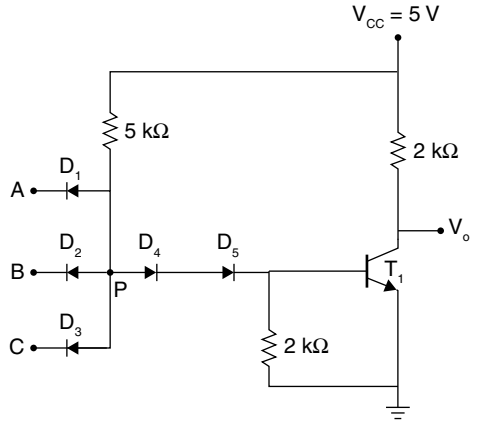


Fig.7.20 DTL NAND gate

The diodes are said to be conducting if they have 0.2 V for low level and between 4 and 5 V for high level. If the input at A is 0.2 V then the diode D_1 conducts and diode current reaches V_{CC} and finally input. The voltage at point P will be the sum of collector–emitter voltage and voltage drop which is assumed to be 0.75 V. Therefore,

$$V_p = V_{CE} + V_D = 0.2 + 0.75 = 0.95$$

This voltage is less than the voltage that is needed for the transistor to conduct. Therefore, the transistor is cut-off and the output voltage is high at +5 V. On the other hand, if the input is high the transistor is in the saturation state which inturn reverse biases the didoes and is off. The output drops to 0.2 V. The DTL has a power dissipation of 12 mW and propagation delay of 30 ns. To increase the fan out one of the diode is replaced with a transistor. The circuit for the modified DTL is shown in Figure 7.21.

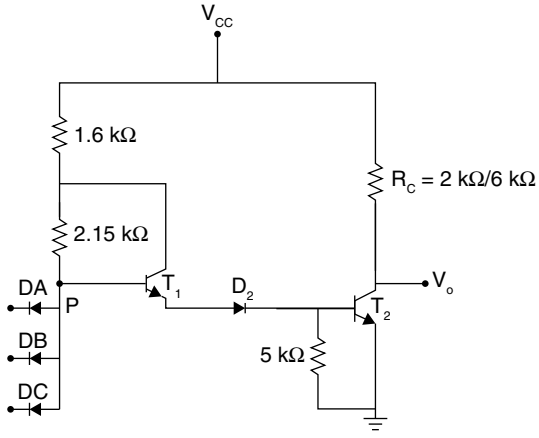


Fig.7.21 Modified DTL NAND gate

When the transistor T_1 is in saturation state then T_2 is cut-off. There are two IC versions of the DTL gate available. One with $R_C = 2$ k and the other is with $R_C = 6$ k. (Low power version). Increase in collector resistance R_C large increases the propagation delay of the system.

Transistor–Transistor Logic (TTL)

Transistor–transistor logic is the most popular and widely used IC logic family. It is also known as T^2L family and was introduced by Texas Instruments in 1964. The circuit of basic TTL gate is shown in Figure 7.22. Let us first consider the working of the basic TTL gate. When the input voltage is high, then transistor T_1 will be in the reverse active region. The base emitter voltage is less and the base collector voltage, V_{BE}, will be more than zero. The current from V_{CC} will flow across the base resistor R_B. It later passes through the collector base junction of T_1 and into the base of transistor T_3. Transistor T_3 is saturated and the output of the gate is 0.2 V.

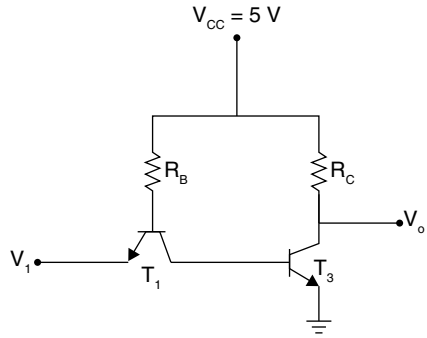


Fig.7.22 Basic TTL gate

When the input voltage is low, then the base emitter voltage V_{BE} is more than zero thereby making the current to flow through the base resistor to the emitter–base junction of T_1 and then to ground. The charge stored at the base of the transistor T_3 does not leak through the resistor instead flows across forward biased collector–base junction of transistor T_1. This drives the transistor T_1 to cut-off. Now the output voltage is high approximately +5 V.

The main difference between the DTL and TTL logic families lies in the operational speed. The transition speed from logic high to low depends on the speed at which the charge is dissipated. The charge stored in the base has a higher discharging current compared to DTL gate. Thus, the TTL is much faster than DTL.

Multi-Emitter TTL Gate: Consider a 3-input TTL NAND gate with a simple output stage. The inputs are applied across emitters of a multiple-emitter transistor. The circuit is shown in Figure 7.23. When all the inputs are held at high it causes the base-emitter junctions of the multiple-emitter transistor to be reverse-biased. A small collector current due to the reverse biasing of the transistor is available at

the inputs. The positive voltage from V_{CC} flows through the base resistor R_B into the base of the transistor T_1. This current passes through the base-emitter junction of the output transistor. Thus, the output transistor conducts and the output voltage low or logic 0.

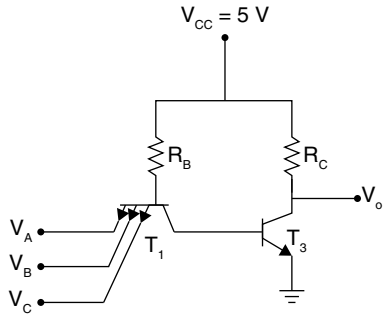


Fig.7.23 3-input TTL NAND gate

When one of the inputs to the multi emitter transistor T_1 is at logic 0, then no current flows through the base of the output transistor and the output voltage is high. This is because the base collector junction of transistor T_1 and the base-emitter junction of the output transistor are connected in parallel across the resistors and ground. When two such stable voltage components having different threshold voltages are connected in parallel, then the current flows across the path with less threshold voltage. This is known as current steering. The same phenomenon occurs when one of the inputs is given logic 0. No current flow is observed at base of transistor T_3 thereby making the transistor to reach cut-off state. The output voltage is high, i.e., logic 1.

Disadvantage of Single Output Stage TTL: The output resistance is extremely high when the output is logic 1. Also this is completely determined by the output collector resistor. The high output resistance restricts the number of inputs that can be connected.

Totem Pole (TTL): The main disadvantage of TTL is the high output resistance of the single output stage. This is overcome using the "totem-pole" TTL gate as shown in Figure 7.24. It consists of the two n-p-n transistors T_3 and T_4, the diode D_1 and the current-limiting resistor R_3. The gate is driven by applying the same current steering idea as above.

When the transistor T_2 is off then this makes transistor T_4 also to be in off state. Transistor T_3 operates in active region and produces high output voltage. When the transistor T_2 is on then this makes transistor T_4 also to be in on state. This causes a low voltage at the output. It is evident from the circuit that the base emitter of T_3 and diode is connected in series. To this series connection, the collector–emitter junction of transistors T_2 and T_4 are connected. When the base current of transistor T_3 is low it drives the transistor to cut-off state.

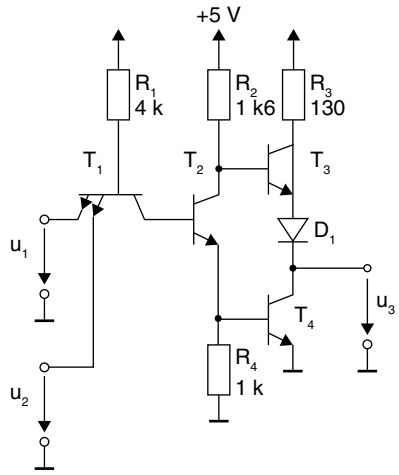


Fig.7.24 Totem pole (TTL)

The resistor R_3 controls the flow of current through the transistor T_3, diode D_1 and transistor T_4 that are connected in series when all are in the conducting state. The output current is also controlled. The resistor R_3 decreases the output resistance as it is connected to the collector of transistor T_3. Thus, these resistors also known as pull-up and pull-down resistors can be removed from the output stage to increase the strength of the gate without affecting the power consumption. The main advantage of TTL with a "TOTEM-pole" output stage is the low output resistance at output logical "1".

The TTL has many subfamilies having standardized labeling system starting with 54 or 74. Some of the subfamilies of TTL are shown in Table 7.1.

Table 7.1

Family	Characteristics	Switching speed	Power consumption	Example
L	Low-power TTL	33ns	1 mW	74L00
H	High-speed TTL	6ns	22 mW	74H00
S	Schottky TTL	3ns	19 mW	74S00
LS	Low-power Schottky TTL	9.5ns	2 mW	74LS00
F	Fast Schottky TTL	>9.5ns	10 pJ	74F00
AS	Advanced-Schottky	>FSTTL	4 pJ	74AS00

Emitter-Coupled Logic

ECL has been around since the early 1960s. The transistor logic has a major drawback of speed constraint that is overcome in emitter-coupled logic (ECL). Emitter-coupled logic is used only when necessary for its high speed. In ECL, the transistors are neither

cut-off nor in the active region. When the transistors are in the active region, the charge stored in the base region of the transistors is kept to a minimum. This allows short duration turn-off. The propagation delay of emitter-coupled logic varies from 1ns to 0.5 ns. The circuit of an emitter-coupled pair is shown in Figure 7.25.

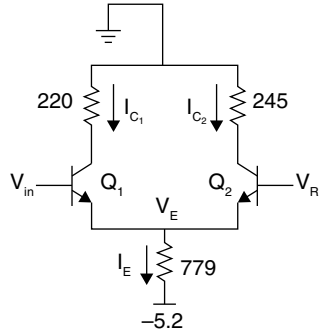


Fig.7.25 Emitter-coupled pair

The emitters of the two transistors are connected together. The base of second transistor is connected to a reference voltage (V_R) and the input voltage is applied to base of transistor Q_1. When the input voltage is LOW, then the transistor Q_1 is cut-off and Q_2 will be conducting and the emitter voltage will be:

$$V_E = V_R - V_{BE} = V_R - 0.75$$

The input voltage can be increased to 0.65 yet maintain Q_1 in OFF state.

$$V_{in}\ \text{max} = V_E + V_{BE\gamma} = V_R - 0.75 + 0.65 = V_R - 0.10 \text{ volts}$$

When the input voltage is higher than this value then Q_1 is turned ON. This causes an increase in the current thereby raising the voltage at the emitters. This voltage drives Q_2 to OFF state. The input voltage rises only a little bit before Q_2 turns OFF entirely. The minimum high level input voltage occurs when

$$V_E = V_R - 0.65$$
$$V_{in}\ \text{min} = V_E + 0.75 = V_R + 0.10 \text{ volts}$$

A complete reversal is caused by a 200 mV swing at the input that makes the transistor ON and OFF.

There are several disadvantages associated with ECL. They use negative power supply that makes it incompatible with any other logic family causing inconvenience in analysis and measurement. ECL requires large currents and small noise margins. They have stable power supply currents when compared to TTL switches thereby reducing noise on the power leads.

Complementary Metal-Oxide Semiconductor

CMOS was introduced in the 1970s and today CMOS is the most widely used logic family. In the name CMOS, "C" stands for complementary as there are N-type and

P-type transistors embedded in them. Metal-oxide semiconductor logic uses CMOS field effect transistors to perform the required switching operations. Electrons are the current carriers in N-type transistors and for P-type transistors holes are the current carriers. The N-type channel is built directly on P substrate with source and drain junctions drawn from N-type doped material. The gate is normally a N-doped conductor. Since it is N-type it requires positive voltage to be applied to gate and drain (with respect to source) for it to conduct and initiate electron flow. This causes current to flow from source to drain. The P-channel device is built on a well which is formed when a deep N-type junction is diffused into the P-substrate. The source and drain junctions are made of P-type doped. Gate is either made of N or P material. The operation of P-type channel requires negative voltage to be applied across gate and drain that results flow of electrons as current from drain to source. The P-channel and N-channel are shown in Figure 7.26.

"M" stands for metal as the gate of the transistor is made of aluminium metal earlier but is replaced by polysilicon. "O" stands for oxide since CMOS devices used oxide-isolated silicon dioxide material between the gate and the channel. "S" refers to the semiconductor material, silicon, a type IV element in the periodic chart.

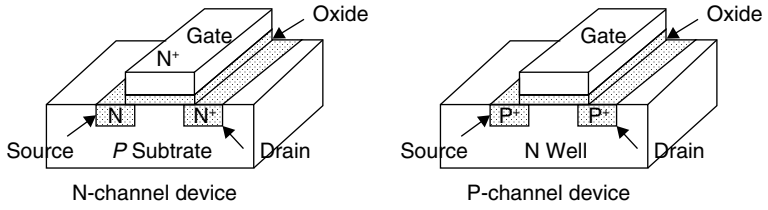


Fig.7.26 P-channel and N-channel CMOS

The gate terminals of both *n*-MOS and *p*-MOS are connected together to form the input terminal and the drain terminals of both *n*-MOS and *p*-MOS are connected together to form the output. It is to be noted that when the input or the gate to source terminal is at LOW, the P-channel FET is ON and when the input or the gate to source terminal is at HIGH, the N-channel FET is ON. When the input voltage is "0", both P-channel FET and N-channel FET goes to OFF state. CMOS provides high input impedance of the order $10^{12}\,\Omega$. The terminals of CMPS are similar to that of a transistor. The gate is like the base, the drain like the collector and the source is like the emitter. There is a major difference between the transistor and CMOS. The controlling parameter in a transistor is current while for FET it is voltage.

Consider the CMOS inverter; the circuit shown in Figure 7.27 has two field effect transistors. One is N-channel field effect transistor and the other is P-channel field effect transistor. When the input or gate voltage is LOW, the P-channel field effect transistors is ON and the N-channel field effect transistor is OFF. When the output is HIGH, the N-channel FET is ON and conducts. The output is always maintained at +5 V supply or ground. The output is always the complement of the input. Thus, the circuit satisfies the function of NOT gate.

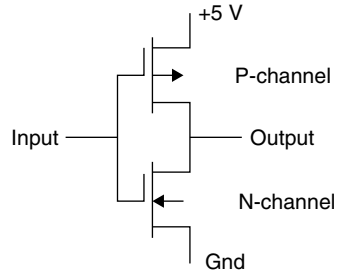


Fig.7.27 CMOS inverter

Consider the working of a 2-input CMOS NAND gate. The circuit is shown in Figure 7.28. It comprises of two P-channel MOSFET and two N-channel MOSFET. The transistors T_1, T_2 are P-type MOSFET and T_3, T_4 are N-channel MOSFET. When both the inputs are high, transistors T_1 and T_2 are cut-off while T_3 and T_4 are conduct. Thus, the output is 0.

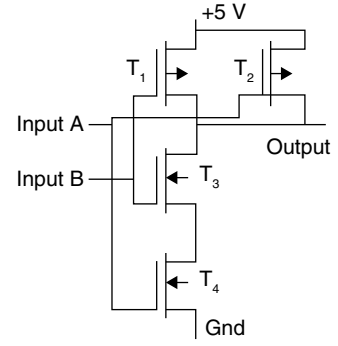


Fig.7.28 CMOS NAND gate

When the input A is high and B is low, transistor T_4 is cut-off and transistor T_2 is ON. Thus, the output is high Now when the input B is high and A is low, transistor T_1 conducts while transistor T_3 is cut-off. When both the inputs are high, then transistors T_1 and T_2 are ON while T_3 and T_4 are OFF. The output in this case is high. Thus, the circuit satisfies the NAND gate.

A CMOS NAND gate has lower impedance compared to an inverter. To avoid high impedance levels, the number of FETs put in series is to be minimised.

Consider a 2-input CMOS NOR gate whose circuit is shown in Figure 7.29, it comprises of four MOSFETs: T_1, T_2, T_3 and T_4. MOSFET T_1, T_2 are P-channel while T_3, T_4 are N-channel. When the inputs A and B are both low then the MOSFET T_1 and T_2 are ON while T_3 and T_4 are OFF. This causes the output equal to the drain voltage, i.e., high. When A is low and B is high: a low in input A causes the MOSFET T_1 is ON while MOSFET T_3 is OFF while high in input B causes the MOSFET T_4 is ON while MOSFET T_2 is OFF. Thus, the output is low level "0" state. Similarly, when A is high the MOSFET T_1 is OFF while MOSFET T_3 is ON while high in input B causes the

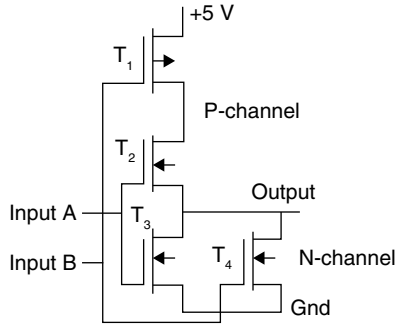


Fig.7.29 CMOS NOR gate

MOSFET T_4 is OFF and T_2 is driven to ON. Thus, the output is again in low level "0" state. When both A and B are high, then the MOSFET do not conduct and thus satisfy the conditions of a NOR gate.

CMOS has very high input impedance. These inputs when left floating can cause high static charges that exceed the damage levels thus destroying the device. This is prevented by providing protection to CMOS gates that has two diodes and a resistor as shown in Figure 7.30. The diodes can be reduced for a low voltage CMOS from two diodes to one diode. The resistors have typical values from 150 Ω to 1 kΩ. The two diodes are connected in parallel and the bottom diode is normally a diffused diode.

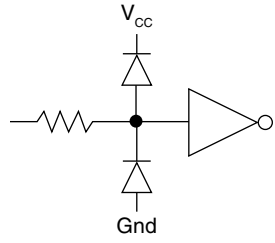


Fig.7.30 CMOS protection

The connection produces SCR across the power supply. The SCR can fuse out when the input voltage exceeds the ground and power supply levels. A latch is created and can be removed only by turning OFF the power supply. The diodes are a protection circuit that prevents latch up.

For proper operation of CMOS, all unused inputs of CMOS should either be connected to the positive supply or ground. This can avoid any damage or spurious operation due to the input floating HIGH or LOW.

Comparison of TTL and CMOS Logic Families

There are number of logic technologies that can be implemented to perform specific desired operations. Table 7.2 summarizes the various characteristics, advantages and disadvantages of the two most commonly used logic families. These make the CMOS a more reliable technology than the TTL.

Table 7.2 Comparison between TTL and CMOS

Characteristic	TTL	CMOS
Fan-in	8	8
Fan-out	10	20–50
Propagation time (nS)	5–15	50–70
Noise margin	0.4 V	0.4 V
Power dissipation	High	Low
Operation speed	Faster	Fast
Switching operation	Poor	Good
Cost	Low	Moderate
Advantages	High fan out, low cost	High fan out, less power dissipation
Disadvantages	Low tolerance level	Low speed
Technology enhancement	Slow	Faster

IC PACKAGING

The integrated chips are packaged to protect them from mechanical damage and chemical contamination. The material used is generally moulded plastic but epoxy, resin or silicone are also used. Ceramic is used when high thermal dissipation capabilities are required. Some special cases require metal or glass covering. The main advantage over packaging the ICs is that it is large enough for electrical connections to be made. It can provide a completed unit large enough to handle, e.g., system on chip (SOC). Three most common packages for ICs are dual-in-line (DIPS), flat pack and axial lead as shown in Figure 7.31. The dual in line is the most commonly used. These packaging are elaborately dealt in chapter.

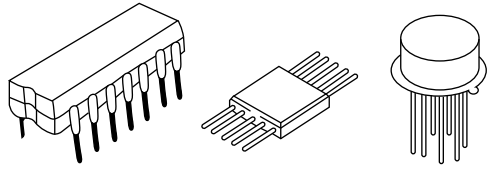


Fig.7.31 IC packaging (a) Dual-in-line (DIPS), (b) Flat pack and (c) Axial lead

TROUBLESHOOTING IN DIGITAL DEVICES

Digital or analog devices are prone to circuit malfunctioning. It is essential to identify the error and repair it for the smooth functioning of the device. The devices can have error due to defect in interconnections, shorted lines, faulty components, etc., time

taken in identifying the error is of greater concern. Many detecting tools for trouble-shooting are available.

Logic Pulser

The logic pulser is a testing tool that dominates the naturally occurring logic state in the circuit. The logic pulser has a micro switch that controls its operation. When the micro switch is pressed then a particular point in the circuit that has logic "1" is driven by the pulser to logic "0" for a very short period. When the micro switch is released then a point having at "0" will drive it to a "1" for a short period. Logic pulser senses the existing voltage level at the node or point in the circuit and produces a voltage pulse in the opposite direction. If the point is LOW, then it produces a narrow positive-going pulse and if the point is HIGH the pulser produces a narrow negative-going pulse. The output impedance of logic pulser is very low of the order 2 Ω or less. This enables it to change the NAND gate output thereby changing the voltage at the point.

When the point is shorted directly to ground terminal or power supply, the logic pulser cannot produce.

Logic Probe

A logic probe is a valuable test instrument for diagnosing errors in digital circuits. The logic probe can be used to detect logic levels but its adaptability increases if it functions as a logic pulser also. The logic probe only detects the occurrence of logic states but when it is used with a logic pulser it can trouble shoot the error in the circuit. The probe has a double LED display. When the red LED glows it is logic "1" and when green LED glows it is logic 0. The logic pulser and probe is used to test a circuit in the following manner. A pulse is given to the circuit by manually operating the micro switch of the logic pulser. It is to be noted that while applying the pulse to the circuit node, the output of NAND gate is not disconnected. To detect shorted nodes in the circuit, the micro switch of the pulser is pressed and both pulser and probe are placed at the same node. When the pulse is at constant LOW and LED display of the probe does not flash then the particular node is shorted to ground as shown in Figure 7.32.

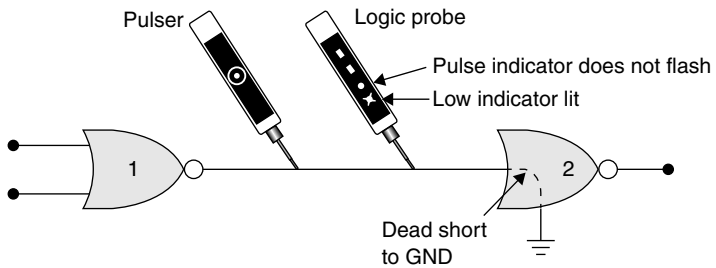


Fig.7.32 Dead short to ground

When the pulse is high and the LED display does not glow then the node is shorted to V_{CC} as shown in Figure 7.33.

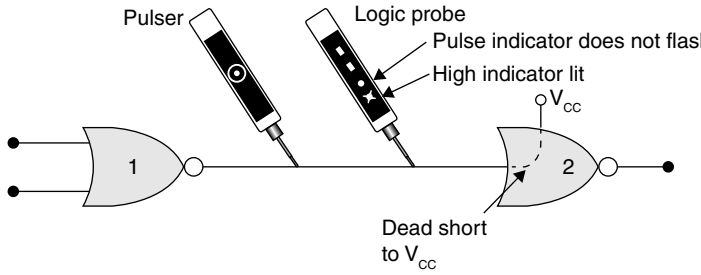


Fig.7.33 Dead short to power supply

SUMMARY

Logic level deterioration caused by diodes restricts their usage as logic gates. On the other hand, the field effect transistors and bipolar junction transistors offer a better option for logic technologies. Although the BJTs and CMOSFET have high fan out, less power dissipation they have few setbacks like propagation delay. Various logic families like RTL, DTL, TTL, ECL and CMOS are available for the designer to execute logic operations. Each logic family has its own characteristics, advantages and disadvantages and it is for the designer to choose the most suitable family among them.

MULTIPLE CHOICE QUESTIONS

1. The process of adding impurity to intrinsic semiconductor is known as
   (a) Doping (b) Metallization
   (c) Etching (d) Masking
2. Integrated circuits use
   (a) Analog inputs (b) Digital inputs
   (c) Both Analog and digital inputs (d) None of the above
3. The very large scale integration (VLSI) technology integrates _____ transistors
   (a) 10000–1 million (b) 100–1000
   (c) 1000–10000 (d) 1 million–10 million
4. Which of the logic families were the first to be integrated?
   (a) Emitter coupled logic (b) Resistor transistor logic
   (c) Diode logic (d) CMOS

5. Which of the logic family uses FET as its switching element?
   (a) TTL (b) Diode logic
   (c) CMOS (d) None of the above
6. The logic family that has the fastest switching action is
   (a) TTL (b) ECL
   (c) CMOS (d) DTL
7. The output is complementary of the input in
   (a) ECL (b) DTL
   (c) RTL (d) CMOS
8. Forward biased diodes are used as switching elements in
   (a) RTL (b) DL
   (c) CMOS (d) ECL
9. Resistors and transistors are used to represent
   (a) DTL (b) CMOS
   (c) TTL (d) RTL
10. The lower the maximum frequency, _____ the propagation delay.
    (a) greater (b) lesser
    (c) does not affect (d) none of the above
11. The CMOS logic family has
    (a) low propagation time (b) high power dissipation
    (c) low fan in (d) high fan out
12. The source and drain regions of a MOSFET are fabricated by the process of
    (a) Oxidation (b) Ion implantation
    (c) Sputtering (d) Polysilicon deposition
13. A transistor with multi input emitter is used in
    (a) DTL (b) CMOS
    (c) TTL (d) ECL
14. Diode is forward biased when _____ voltage is applied to anode and _____ voltage to cathode.
    (a) Positive, negative (b) Negative, positive
    (c) Both are applied zero volts (d) Both are applied +5 V
15. Most commonly used TTL logic is
    (a) Schottky TTL (b) Tristate TTL
    (c) Low-power TTL (d) Open-collector TTL
16. Which of the combination form the terminals of field effect transistor?
    (a) Emitter, Drain, Gate (b) Source, Drain, Gate
    (c) Emitter, Base, Collector (d) None of the above

17. Large scale integration circuits use
    (a) TTL (b) DL
    (c) CMOS (d) DTL
18. Power dissipation is minimum in
    (a) CMOS (b) TTL
    (c) RTL (d) DTL
19. Schottky TTL gates have propagation delay time of the order of
    (a) 6 ns (b) 5 ns
    (c) 2 ns (d) 8 ns

Answers

1. (a) 2. (c) 3. (a) 4. (b) 5. (c) 6. (b) 7. (a)
8. (b) 9. (d) 10. (a) 11. (d) 12. (d) 13. (c) 14. (a)
15. (a) 16. (b) 17. (c) 18. (a) 19. (c)

DESCRIPTIVE QUESTIONS

1. Explain the classification of logic families.
2. Explain the fabrication and working of field effect transistors.
3. Discuss the important characteristics of digital ICs.
4. Distinguish between discrete and integrated circuits.
5. Write notes on:
   (a) Fan-out
   (b) Fan-in
   (c) Power dissipation
   (d) Propagation delay
   (e) Noise margin
6. Draw the circuit for DL AND gate and DL OR gate. Explain its operation.
7. Discuss totem pole output (TTL) with necessary circuit. What are its merits and demerits?
8. Explain the working of RTL NAND gate.
9. Explain the basic circuit of emitter-coupled logic.
10. Write note on logic pulser.
11. Write note on logic probe.
12. Classification of IC based on size.
13. Explain the working of CMOS as NOT, NAND and NOR gates with suitable circuit diagrams.
14. Explain the working of DTL NAND gate.
15. Write notes on various tools available for troubleshooting in digital circuits.

COMBINATIONAL LOGIC CIRCUITS

COMBINATIONAL CIRCUITS

Digital logic circuits can be classified into "combinational" and "sequential." A combinational logic circuit is one whose output solely depends on its current inputs. Sequential circuits, on the other hand, are built using combinational circuit and memory elements called "flip-flops". These circuits generate output which depends on the current and previous states. Some examples of sequential circuits are counters and shift registers. Another major difference between the combinational and sequential circuits is the feedback mechanism. A number of logic gates and inverters are used in the combinational circuit but without feedback loops. The feedback loop is a connection through which a part of the output is fed back to the input and thus performs a sequential operation. We would concentrate on combinational circuits as sequential circuits would be discussed in the later chapters.

A combinational circuit is a logic circuit comprising of logic gates that generate output depending on the inputs applied at a given instant of time. The analysis of a combinational is done by first implementing the logic diagram using logic gates and invertors. Then the Boolean expression and truth table of the circuit is interpreted. Finally the output is derived depending on the inputs in the current states. Certain combinational circuits may have one or more outputs. In this chapter, we will discuss methods that apply to single-output circuits. Some examples of typical combinational circuits are binary adders, subtractors, multiplexers, demultiplexers, decoders, and encoders. On completion of the chapter, the reader would have an in-depth understanding of combinational circuits.

BINARY ADDERS

The binary adder is a combinational circuit that can perform summation of the input binary numbers. We will study and construct the various adder circuits.

HALF ADDER

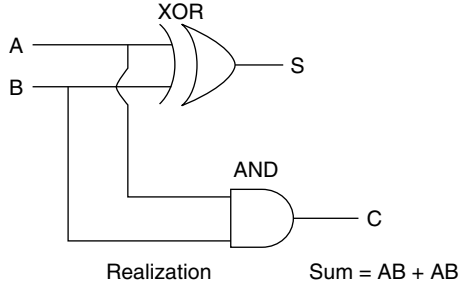
Half adder is the simplest of all adder circuits. Half adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit (S) and carry bit (C) both as the output. Let us consider two inputs bits A and B, then sum bit (S) is the X-OR of A and B and the carry bit (C) will be the AND of A and B. It is evident from the function of a half adder that it requires one X-OR gate and one AND gate for its construction. Let us first consider the addition of single bits.

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 10$$

These are the least possible single-bit combinations. We notice that the output as a result of the addition of 1 + 1 results in 10. This can be solved using an EXOR Gate, or the sum result must be re-written as a 2-bit output. Thus, the above equations can be written as

$$0 + 0 = 00$$
$$0 + 1 = 01$$
$$1 + 0 = 01$$
$$1 + 1 = 10$$

Here the output "1"of "10" becomes the carry-out. The "SUM" is the normal output and "CARRY" is the carry-out. Though the half adder is the simplest adder circuit, it has a major disadvantage. A half adder can add only two input bits (A and B) and is not affected by the carry of the input. As a result, if the input that is given to a half adder has a carry, then it will be neglected and it adds only the bits A and B. But this results in an incomplete binary addition and hence it gets its name a half adder. For complex addition, two or more half adders are connected together forming a full adder. The truth table and XOR//AND understanding of a half adder are shown in Figure 8.1.

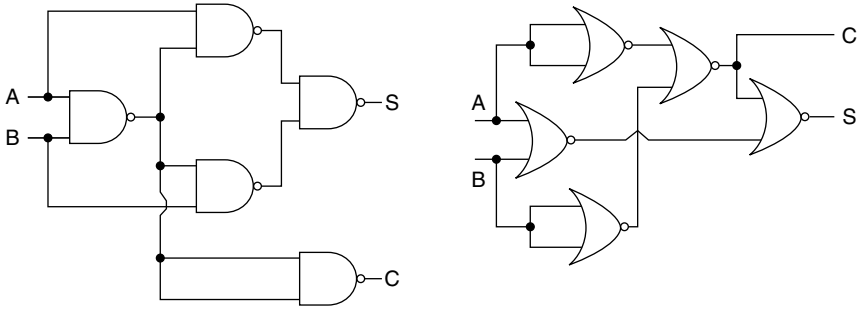


Truth Table

Input		Output	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Fig.8.1

NAND gates or NOR gates can also be used for comprehending the working of the half adder. The relevant circuit diagrams are shown in Figure 8.2.



Half adder using NAND logic Half adder using NOR logic

Fig.8.2

The ANSI/IEEE standard logic symbol for half adder is shown in Figure 8.3.

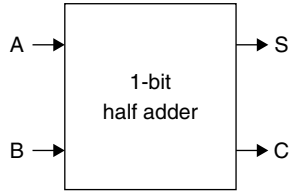


Fig.8.3

FULL ADDER

The full adder is a much complex adder circuit compared to the half adder. The major difference between a half adder and a full adder is the number of input terminals that are fed to the adder circuit. The full adder has three inputs and two outputs. The first two inputs are A and B and the third input is an input carry designated as C_{IN}. The full adder is designed in such a way that it can take in eight bits together to create a byte-wide adder and cascade the carry bit from one adder to the next. The output carry is designated as C_{OUT} and the normal output is designated as S. Consider the following truth table for a full adder.

Truth Table

Inputs			Outputs	
A	B	C_{IN}	C_{OUT}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0

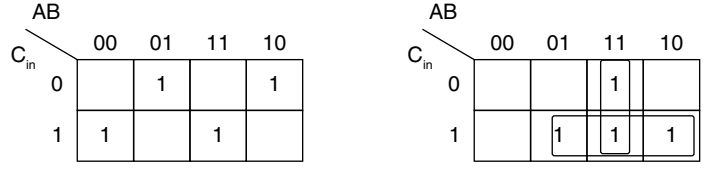
Truth Table (*continued*)

Inputs			Outputs	
A	B	C_{IN}	C_{OUT}	S
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

The output S is an XOR between the input A and the half adder sum output with B and C_{IN} inputs. The C_{OUT} will only be true if any of the two inputs out of the three are HIGH. Thus, a full adder circuit can be implemented with the help of two half adder circuits. The function of the first half adder is to add the inputs A and B to produce a partial sum. The second half adder adds the output of the first adder and the C_{IN} to get the final output (S). If there will be an output carry. Thus C_{OUT} will be an OR function of the half adder carry outputs. The sum of the products (SOP) for the above truth table with sum S_n and carry C_n is given as:

$$S_n = A\overline{B}\,\overline{C}_{in} + \overline{A}B\overline{C}_{in} + AB\,C_{in} + \overline{A}\,\overline{B}C_{in}$$
$$= (A\overline{B} + \overline{A}B)\,C_{in} + (AB + \overline{A}\,\overline{B})\,C_{in}$$
$$= (A \oplus B)\,C_{in} + \overline{(A \oplus B)}$$
$$= A \oplus B \oplus C_{in}$$
$$C = AB\overline{C}_{in} + A\overline{B}C_{in} + \overline{A}BC_{in} + ABC_{in} \qquad \text{using Boolean identities}$$
$$= AB + (A \oplus B)\,C_{in} \qquad A + \overline{A} = 1 \text{ and } A + A = A$$

The Karnaugh map for the truth table is given as:



The full adder circuit is shown in Figure 8.4. It comprises of two XOR gates and two AND gates and one OR gate. It can also be constructed using a NAND gate by employing double complement method.

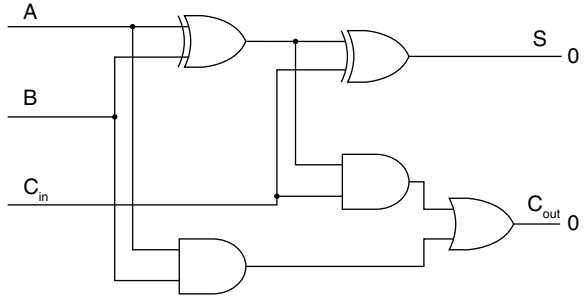


Fig.8.4 Full adder circuit

Though the implementation of larger logic diagrams is possible with the above full adder logic, a simpler symbol is mostly used to represent the operation. The IEEE symbol of a full adder is as shown in Figure 8.5.

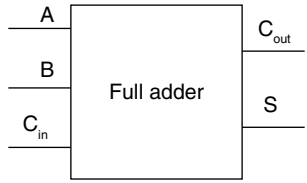


Fig.8.5 Single-bit full adder

The Block diagram of full adder implementation via a pair of half adders is as shown in Figure 8.6.

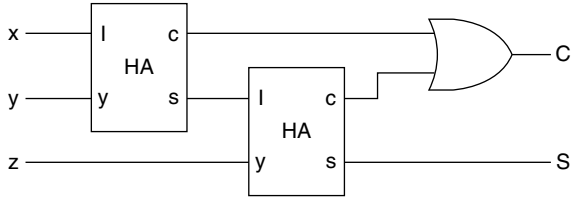


Fig.8.6 Full adder using two HA

Full adder implementation via a pair of half adders, expanded in terms of gates is shown in Figure 8.7. The two half adders are indicated by the dashed boxes.

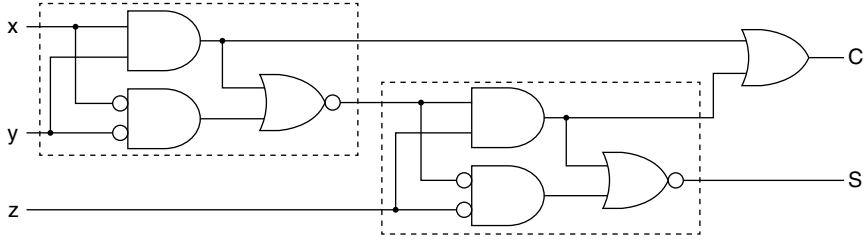


Fig.8.7 Full adder circuit using half adder

The circuit has 14 gate inputs and has the longest path from input to output as it passes through 4 gates. However, while designing a multi-bit binary adder, the possibility of carry propagation is the limiting factor. Hence this design is quite competitive.

ADDITION OF MORE THAN 2 BITS BY FULL ADDER

The above-mentioned adder is used to sum up to 2 bits together taking a carry from the next lower order of magnitude and sending a carry to the next higher order of magnitude. For a multi-bit operation, each bit must be represented by a full adder and all

the bits must be added simultaneously. Thus, to add two 8-bit numbers, we need 8 full adders which can be formed by cascading two of the 4-bit blocks. The addition of two 4-bit numbers is shown below.

The addition of 2 inputs A and B results in a sum S and carry C. The carry C is added to the next unit value of power 2s in the binary. The general addition of 2 bits A_n, B_n involves sum and carry from previous addition C_{n-1}. Addition of the bits can be done in two ways, namely, parallel and serial addition. Serial addition is the mental addition we do by complementing an addition after another with or without carry. Serial additions are simpler and cost effective as they require only one full adder. The parallel addition, on the other hand, involves more numbers. Each bit requires one full adder; if there are 5 bits then we require 5 full adders and in general N bits require N full adders. In parallel addition, the carry output of the first full adder is connected to the carry of the second and is continued in the same manner. The carry input of the first full adder is earthed and hence works as a half adder instead of a full adder. The carry output of the first is the LSB and the carry output of the last full adder is the MSB. The values at the terminal A of all the adders are stored as addend and those values of B's are stored as augend. Similarly, the sum and carry are stored in storage registers. The sum register of the first becomes the addend of the next. Multi-bit addition using full adder is shown in Figure 8.8.

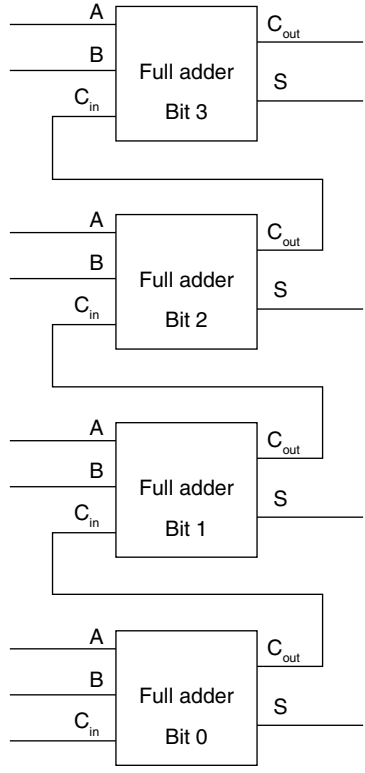


Fig.8.8 Multi-bit addition using FA

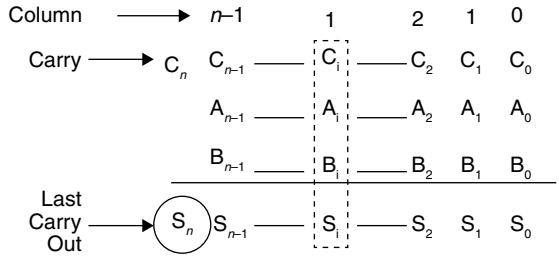
The main disadvantage in a parallel adder is that the carry is passed on to all the adders even if there are no carries in the subsequent adders. This is called a ripple and hence the adder is named ripple carry adder. Also since the adders are TTL circuits, they have a natural delay in itself. If the delay for an adder is 10ns then for a 3 adders the delay will be 30ns. This delay can be overcome by a look-ahead carry adder.

Look-Ahead Carry Adder

The adders produce carry propagation delay while performing other arithmetic operations like multiplication and division as it uses several addition or subtraction steps. This is a major problem of the adder and hence improving the speed of addition will improve the speed of all other arithmetic operations. Hence reducing the carry propagation delay of adders is of great importance. There are different logic design approaches that have been employed to overcome the carry propagation problem. One widely used approach is to employ carry look ahead that solves this problem by calculating the carry signals in advance, based on the input signals. This type of adder circuit is called carry look-ahead adder (CLA adder). Here a carry signal will be generated in two cases:

1. Input bits A_i and B_i are 1
2. When one of the two bits is 1 and the carry-in (carry of the previous stage) is 1

This is understood better by considering the case of adding two *n*-bit numbers, A and B.



Consider a full adder circuit (Figure 8.9) used to add the operand bits in the i^{th} column; namely, A_i and B_i and the carry bit coming from the previous column (C_i).

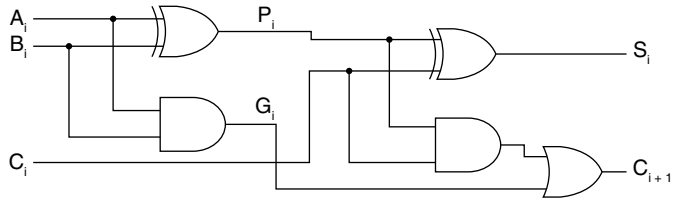


Fig.8.9 Full adder circuit for operand bits addition

In this circuit, the 2 internal signals P_i and G_i are given by:

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

The output sum and carry can be defined as

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

G_i is known as the carry generate signal since a carry (C_{i+1}) is generated whenever G_i =1, regardless of the input carry (C_i). When $P_i = 1$, the input carry is propagated to the output carry, i.e., C_{i+1}. = C_i (note that whenever $P_i = 1$, $G_i = 0$). P_i is known as the carry propagate signal. P_i and G_i values depend on the input bits A_i and B_i. After propagation these signals settle to their steady-state value.

The Boolean expression of the carry outputs of various stages can be written as follows:

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0)$$
$$= G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 C_0 + P_2 P_1 P_0 C_0$$

$$C_4 = C_3 + P_3 C_3$$
$$= G_3 + P_3 G_3 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$$

Therefore, i^{th} term of the carry output is expressed in the form $C_i = F_i$(P's, G's, C_0). In other words, each carry signal is expressed as a direct SOP function of C_0 rather than its preceding carry signal. Since the Boolean expression for each output carry is expressed in SOP form, it can be implemented in 2-level circuits. The 2-level implementation of the carry signals has a propagation delay of 2 gates, i.e., 2τ. The 4-bit carry look-ahead (CLA) adder consists of 3 levels of logic shown in Figure 8.10.

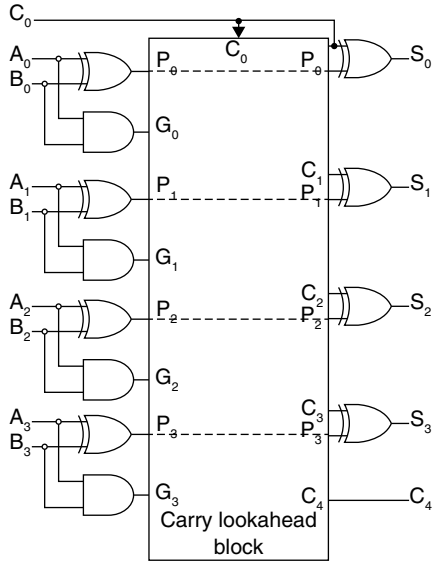


Fig.8.10 CLA adder

First Level: All P and G signals are generated in this full adder. Each P and G logic consists of an XOR gate and an AND gate. Therefore, a total of four sets are present. The output signals from this level are produced after 1τ.

Second Level: This level has the carry look-Ahead (CLA) logic block which consists of four 2-level implementation logic circuits. It generates the carry signals (C_1, C_2, C_3, and C_4) as defined by the above expressions. Output signals of this level (C_1, C_2, C_3, and C_4) will be valid after 3τ.

Third Level: Here four XOR gates are present which generate the sum signals ($S_i = P_i \oplus C_i$). Output signals of this level (S_0, S_1, S_2, and S_3) will be valid after 4τ. Thus, the 4 sum signals (S_0, S_1, S_2, and S_3) will all be valid after a total delay of 4τ compared to a delay of $(2n + 1)\tau$ for ripple carry adders.

For a 4-bit adder ($n = 4$), the ripple carry adder delay is 9τ. The disadvantage of the CLA adders is that the carry expressions are complex when the number of bits keep increasing. Thus, CLA adders are implemented for a maximum of 4 bits.

PARALLEL ADDER–SUBRACTOR

Adder–subtractor circuit performs addition and subtraction operations on the input bits. Figure 8.11 shows the logic diagram of a 4-bit adder–subtractor circuit. It has a mode control signal M which determines if the circuit is to operate as an adder or a subtractor.

The input to the XOR gate is from input M and one of the inputs of B, i.e., B_i. When one of the inputs of XOR gate is 0 then the output of XOR will be same as the second input. On the other hand, when one input of XOR gate is 1 then the output of XOR will be complement of the second input.

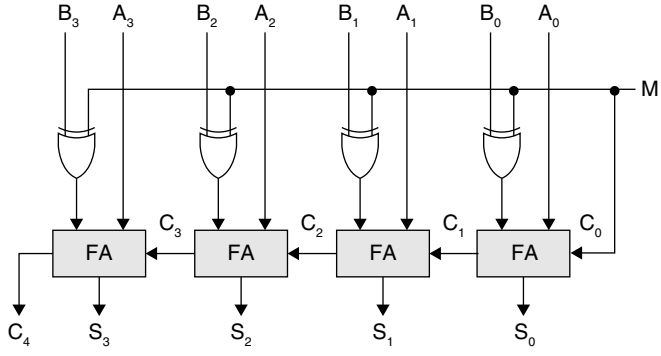


Fig.8.11 Adder–Subractor circuit

When the mode control is at logic LOW, then the output of XOR gate will be $B_i \oplus 0 = B_i$. Then the full adders receive the value of B, and the input carry C_0 is 0, the circuit performs A plus B. When the mode control is at logic HIGH, then the output of XOR gate will be $B_i \oplus 1 = B_i$'. If the full adders receive the value of B', and the input carry C_0 is 1, the circuit performs A plus 1's complement of B plus 1, which is equal to A minus B.

Binary Multiplier

Multiplication of binary numbers is performed in the same way as with decimal numbers. Starting with the least significant bit of the multiplier, each bit is multiplied with the multiplicand. This results in a partial product. When the second resultant product occurs, the previous partial product is shifted one bit to the left. Successive partial products are moved in the similar manner. Final product is obtained by adding these shifted partial products.

Example 8.1 Consider multiplication of 2-bit numbers "A" and "B". The circuit is as shown in Figure 8.12. Let the product be C, such that

$$C = A \cdot B$$

First the least significant bits of number A is multiplied to the product of B_1B_0. This gives the first partial product. This partial product can be obtained using AND gates because the multiplication of two bit, such as A_0 and B_0 produces a 1 if both bits are 1 and in all other cases it produces a 0 similar to AND operation. The second partial product is formed by multiplying the B_1B_0 by A_1 and is shifted one position to the left. Then the resultant two partial products are added with two half adders (HA). When the number of bits is more than a half adder is replaced with a full adder. It is to be noted that the least significant bit of the product does not enter the adder, since it is formed by the output of the first AND gate as shown below. A binary multiplier with more bits can be constructed in a similar manner.

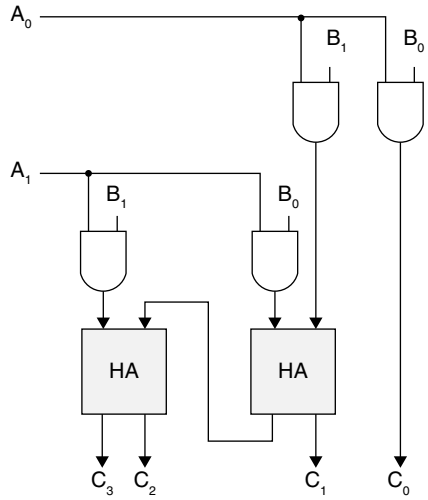


Fig.8.12

MULTIPLEXERS

Multiplexers are combinational circuits that can select one desired input signals from among n inputs and passes it to the single output line. These are also called data selectors. They have 2^n select lines, where n is the select lines or selector. Selection of

the particular input to be passed to the output is controlled select inputs. The block diagram of a 4 × 1 multiplexer is shown in Figure 8.13.

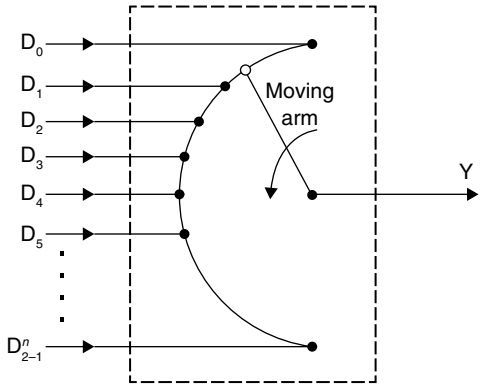


Fig.8.13 4 × 1 multiplexer

2 × 1 MULTIPLEXER

Consider a 2 × 1 MULTIPLEXER, it has two input lines D_0 and D_1, one select input S, and one output line Y. The value assigned to S determines the output Y. The circuit is shown in Figure 8.14.

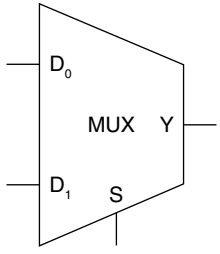


Fig.8.14 2 × 1 multiplexer

When S = 0, then $Y = D_0$

S = 1, then $Y = D_1$

Thus, the output signal is expressed as:

$$Y = \overline{S}D_0 + S\overline{D}_1$$

4 × 1 MULTIPLEXER

A 4 × 1 multiplexer as shown in Figure 8.15, has four input lines D_0, D_1, D_2, D_3, two select inputs S_0 and S_1, and one output terminal Y.

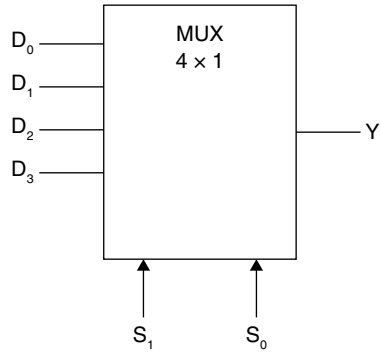


Fig.8.15 4 × 1 multiplexer

The output here depends on the values taken by S_0 and S_1 which are represented in Table 8.1.

Table 8.1

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Thus, the output signal Y can be expressed as:

$$Y = \underbrace{\overline{S}_1\overline{S}_0}_{\text{minterm}}D_0 + \underbrace{\overline{S}_1\overline{S}_0}_{\text{minterm}}D_1 + \underbrace{\overline{S}_1\overline{S}_0}_{\text{minterm}}D_2 + \underbrace{\overline{S}_1\overline{S}_0}_{\text{minterm}}D_3$$

$$m_0 \qquad m_1 \qquad m_2 \qquad m_3$$

Therefore, in general, if the multiplexers have n select inputs, the output Y is given by:

$$Y = m_0D_0 + m_1D_1 + \ldots + m_{2^n-1}D_{2^n-1}$$

Where $m_i = i^{th}$ minterm of the select inputs thus,

$$Y = \sum_{i=0}^{2^n-1} m_iD_i$$

QUAD 2 × 1 MULTIPLEXER

Quad 2 × 1 multiplexers are used when we need to select any one of the two 4-bit numbers A and B based on some select signal S. The output Y is also a 4-bit number. The circuit is as shown in Figure 8.16. The 4-bit output number Y will be equal to number "A" if the select line S = 0. When the select line S = 1 then the output is "B".

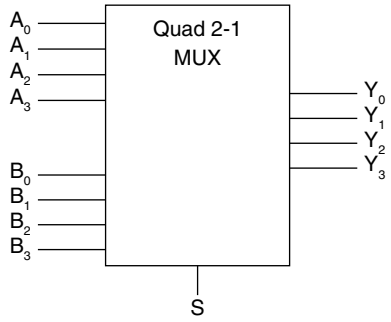


Fig.8.16 Quad 2 × 1 multiplexer

The circuit can be constructed using four 2 × 1 multiplexers, where the output of each of the multiplexers gives one of the outputs Y_i.

Large multiplexers can be constructed by combining two or more small multiplexers. An 16 × 1 multiplexer can be constructed by using two (8 × 1 multiplexer) and one (2 × 1 multiplexer) as shown in Figure 8.17.

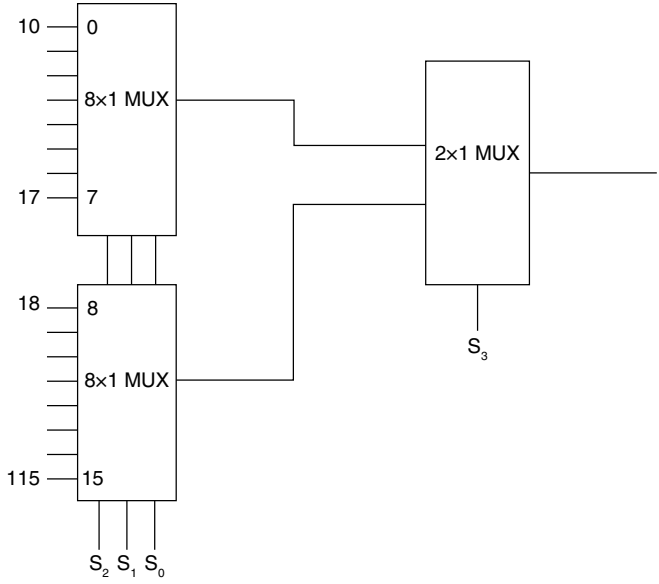


Fig.8.17 16 × 1 multiplexer

IMPLEMENTATION OF BOOLEAN FUNCTIONS USING MULTIPLEXERS

The primary function of multiplexers is its application as data selectors. Apart from working as data selectors, the multiplexers can be used in execution of Boolean expressions. A Boolean function of X variables can be executed using a 2^X input multiplexer.

To perform logic functions, the data lines must be assigned logic high (1) or logic low (0). The inputs are applied to the select lines. As the number of variables keeps increasing, the data lines are also used to apply input variables.

Combinational Circuit Implementation using Multiplexers

Consider a function of *n*-variables. The multiplexers can be used to implement this function in two ways. *Firstly*, multiplexers can be chosen such that they have *n*-select inputs. The *n* variables can be connected to the *n* select inputs. The output Y is given as

$$Y = m_0D_0 + m_1D_1 + m_2D_2 + ... + m_{2^n-1}D_{2^n-1}$$

Alternatively,

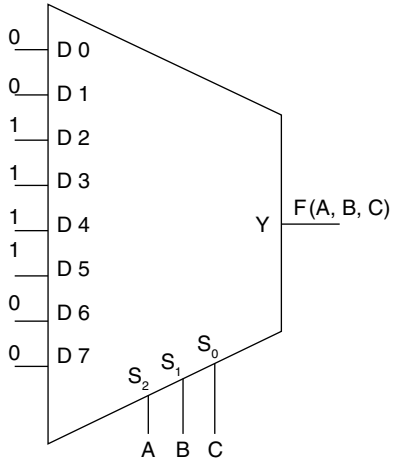
$$Y = \sum_{i=0}^{2^n-1} m_iD_i$$

Where m_i = i^{th} minterm of the select inputs. The output of the multiplexer is the sum of all minterms (m_i) which have their corresponding inputs (D_i) equal to 1. Thus, multiplexer can be used to implement any function of *n*-variables with *n*-select inputs with logic 1 or 0.

Example 8.2 Implement the function $F(A, B, C) = \Sigma(2, 3, 4, 5)$.

There are three variables A, B, C and hence $n = 3$. This requires a multiplexer with 3 select inputs, i.e., an 8×1 multiplexer. Variable A is the most significant select input S_2, while the least significant variable C is connected to the least significant select input S_0, therefore,

$S_2 = A$, $S_1 = B$, and $S_0 = C$

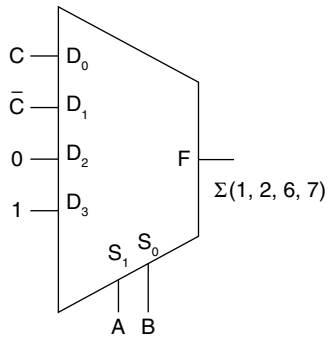


The multiplexer output expression (sum of minterms) includes minterm 1, we assign $D_1 = 2$. Likewise, to include minterms 3, 4, and 5 in the sum of minterms expression, while excluding minterms 0, 1, 6, and 7, the following input (D_i) assignments are made

$$D_1 = D_3 = D_5 = D_6 = 1$$
$$D_0 = D_2 = D_4 = D_7 = 0$$

The second method by which multiplexers can be used to implement any logic function of n variables is by choosing the select inputs as $(n - 1)$. A 3-variable function requires a 4×1 multiplexer. This is accomplished by choosing $n - 1$ variables to be connected to the multiplexer select lines. Then the truth table of the function is constructed by grouping the $n - 1$ select input variables together. The values of multiplexer input line D_i will be 0, or 1, or nth variable or complement of nth variable in the given function F.

Consider the function $F(A, B, C) = \Sigma(1, 2, 6, 7)$. The implementation of the function requires a 4-to-1 line multiplexer. A is applied to the select line S_1 and B is applied to the select line S_0. This is represented in the figure below.



	A	B	C	F	
1	0	0	0	0	F = C
	0	0	1	1	
2	0	1	0	1	F = C′
	0	1	1	0	
3	1	0	0	0	F = 0
	0	1	1	0	
4	1	1	0	1	F = 1
	1	1	1	1	

Implementing function with MUX with $n - 1$ select inputs

DEMULTIPLEXER

Demultiplexers are those that perform the inverse of the multiplexing operation on a given function. They are abbreviated as DEMUX. It has one input line (E) and transmits it to one of 2^n possible output lines (D_0, D_1, D_2 …, $D_{2^n - 1}$). The specific output is selected based on the bit combination of n select inputs. The basic diagram of a DEMUX is shown in Figure 8.18.

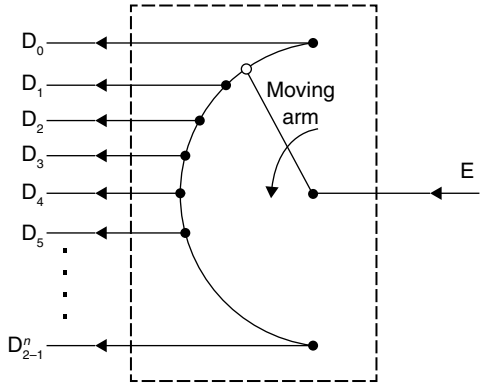


Fig.8.18 DEMUX block diagram

The basic design and working of a DEMUX can be understood from the following example. Consider a 1-to-4 line demultiplexer. It has two select lines, S_1 and S_0. The input E is applied to one of the outputs and is specified by the two select lines, S_1 and S_0. When the select lines are given the values as shown then the output is given according to Table 8.2.

Table 8.2 Select lines with corresponding output

S_1	S_0	E	Output function
0	0	D_0	$S_1'S_0'$
0	1	D_1	$S_1'S_0$
1	0	D_2	S_1S_0'
1	1	D_3	S_1S_0

A demultiplexer is identical to a 2-to-4 decoder with enable input (Figure 8.19).

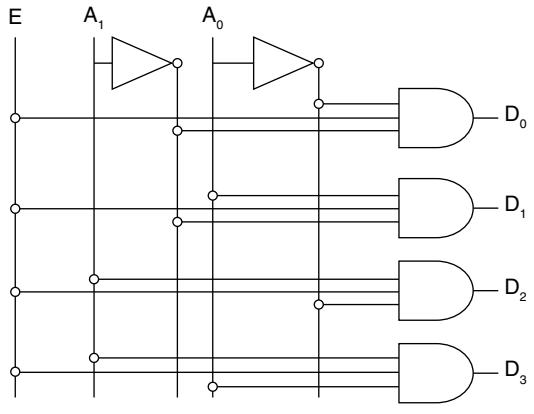


Fig.8.19 2-to-4 decoder with enable input

In case of a decoder, the inputs are A_1 and A_0, and the enable is input E. But in demultiplexer the input E provides the data, while other inputs accept the selection variables. Both the circuits have different applications yet their logic diagrams are exactly the same.

Consider the 74LS or 74HC139 which are examples of 1-to-4 line demultiplexers/decoders. Both the ICs have single enable and two input select pins. There are 4-output pins. When logic HIGH is applied to the enable, then all the four outputs are high regardless of the state of the select pins. When a logic LOW is applied to the enable then the outputs chosen by the address pins A and B will be LOW and other outputs will be HIGH. Consider the truth table of a 1-to-4 line DEMUX as in Table 8.3.

Table 8.3 Truth table for 1-to-4 DEMUX

Decimal value	Enable	Inputs		Outputs			
	E	A_1	A_0	D_0	D_1	D_2	D_3
	0	X	X	0	0	0	0
0	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0
2	1	1	0	0	0	1	0
3	1	1	1	0	0	0	1

Thus, the enable pin defines whether the circuit functions as a demultiplexer or decoder. When the enable pin functions as a data input pin then it is a demultiplexer. When the enable pin is LOW it acts as a decoder. IC 74LS138/74HC138 have 3-enable inputs out of which one pin acts as data input when the chip functions as a demultiplexer as shown in Figure 8.20.

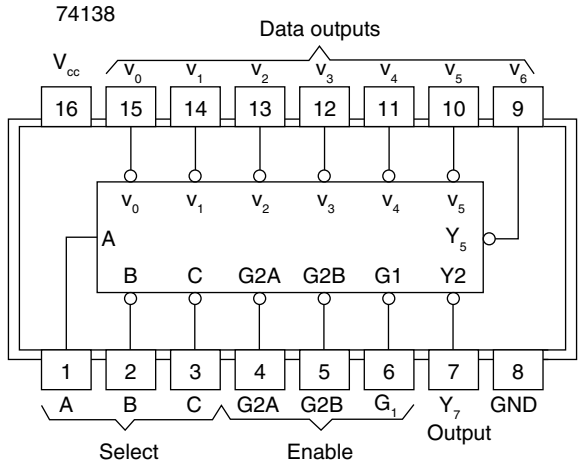


Fig.8.20 IC 74LS138 pin out

CODE CONVERTORS

Code convertors are circuits that are capable of converting the input data existing in a particular code to an output data in a desired code. They are also known as decoders. 74LS42 BCD to decimal decoder, 74LS47 BCD to 7-segment decoder, 74HC42 BCD to decimal convertors are examples of code convertors.

Consider a 4-to-16-line decoder. It has 16 AND gate outputs out of which only 10 outputs ever become 1. These 10 outputs from a 4-to-16-line decoder are used. The truth table for each of seven output functions (the S_i) in terms of the four input variables is shown in Table 8.5. The circuit external to the decoder consists of seven OR gates, one OR gate corresponding to each segment. The outputs from the decoder determines the input to the OR gate. The minterm corresponding to each code word for which that segment output has the value 1 is chosen. The list of minterm for the outputs corresponding to some of the segments is as follows:

$$S_3 = \Sigma(3, 5, 6, 8, 10, 11, 12)$$
$$S_4 = \Sigma(3, 4, 5, 6, 7, 10, 11, 12) \quad (8)$$
$$S_5 = \Sigma(3, 4, 6, 7, 8, 9, 10, 11, 12)$$
$$S_6 = \Sigma(3, 5, 6, 8, 9, 11)$$

Table 8.4 Truth table 4–16 line decoder

Decimal digit	Inputs: Excess-3				Outputs: 7-segment						
	w	x	y	z	S_1	S_2	S_3	S_4	S_5	S_6	S_7
0	0	0	1	1	1	1	1	1	1	1	0
1	0	1	0	0	0	0	0	1	1	0	0
2	0	1	0	1	1	0	1	1	0	1	1
3	0	1	1	0	0	0	1	1	1	1	1
4	0	1	1	1	0	1	0	1	1	0	1
5	1	0	0	0	0	1	1	0	1	1	1
6	1	0	0	1	1	1	0	0	1	1	1
7	1	0	1	0	0	0	1	1	1	0	0
8	1	0	1	1	1	1	1	1	1	1	1
9	1	1	0	0	0	1	1	1	1	0	1

Only one of the OR gates is used. A BCD to 7-segment decoder is shown in Figure 8.21. When a code corresponding to a decimal digit appears at the input, the appropriate segments will light up, displaying the digit.

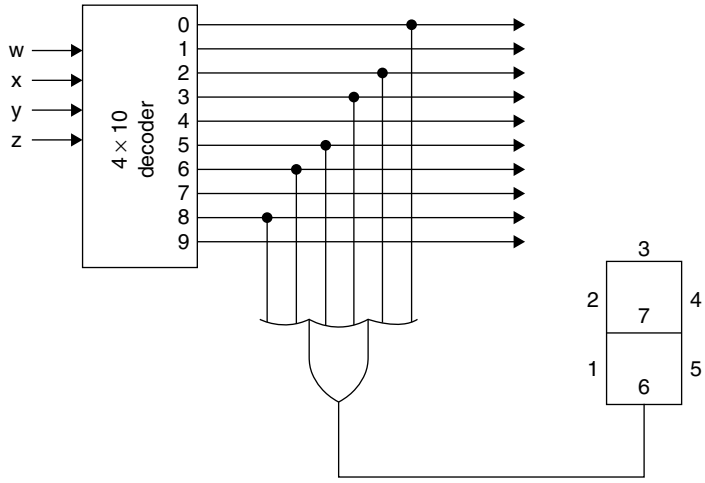


Fig.8.21 BCD to 7-segment decoder

DECODERS

Decoders are combinational circuits that help in converting one form of coded inputs to the other. The well-known examples of decoders are binary n-to-2^n decoders and 7-segment decoders.

The binary decoder has n inputs and 2^n outputs. The output, 2^n, is the maximum value possible as n inputs provides 2^n minterms or maxterms. Therefore, a decoder produces one of the 2^n minterms or maxterms at the outputs based on the input combinations. Consider the 2-to-4 decoder as an example, the block diagram and the truth table of this decoder is shown in Figure 8.22.

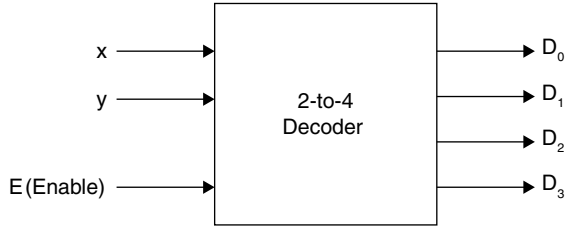


Fig.8.22 2-to-4 decoder

Table 8.5 Truth table of a 2-to-4 decoder

E	x	y	D_0	D_1	D_2	D_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

It is evident from the truth table 8.6 that there is only one active output for the combinations of the inputs. The Boolean expression of the output signals are

$$D_0 = E\,x'y', D_1 = E\,x'y, D_2 = E\,x\,y' \text{ and } D_3 = E\,x\,y$$

The logic diagram for the 2-to-4 decoder obtained from the above function is as in Figure 8.23.

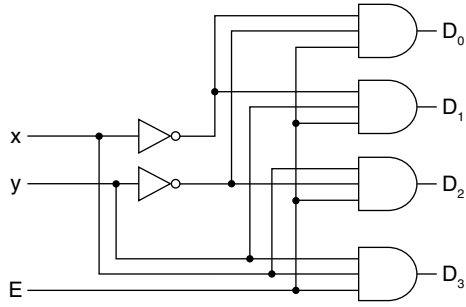


Fig.8.23 Logic diagram of 2-to-4 decoder

This method can be adopted to obtain the logic diagram for any *n*-to-2^n type decoder. Decoders are constructed using NAND gates instead of using AND a gate as it is easy and less expensive to build. IC 74139 is an example of a commercial *n*-to-2^n line decoder. The chip comprises of two 2-to-4 decoders with active low enable for each. They constructed using the NAND gates. The pinout diagram is shown in Figure 8.24.

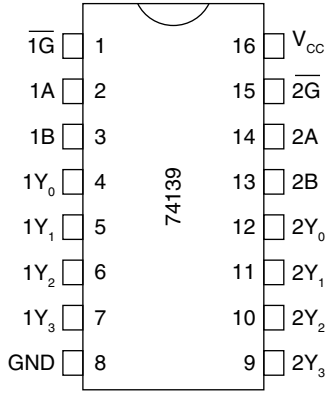


Fig.8.24 74139-point diagram

Decoders with enable inputs can be combined and used to construct higher order decoders. A 3-to-8 decoder can be constructed using two 2-to-4 decoders with enable inputs as shown in Figure 8.25. The two least significant bits, A_1 and A_0, are connected to both 2-to-4 decoders. The most significant bit A_2 is given to the enable input of one decoder and the complement of most significant bit A_2 is connected to the enable of the second decoder. When $A_2 = 0$, upper decoder is enabled and lower decoder is disabled. Thus, making the outputs of the upper decoder correspond to minterms D_0 through D_3. When $A_2 = 1$, upper decoder is disabled and the lower decoder is enabled thereby making the outputs of the lower decoder correspond to minterms D_4 through D_7.

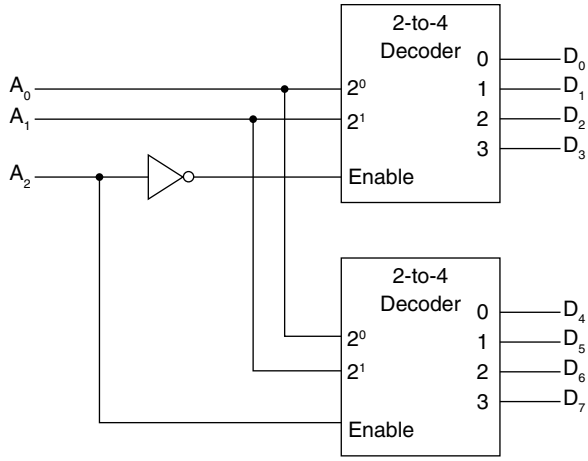


Fig.8.25 3-to-8 decoder circuit

As Boolean functions are expressed as sum of products (minterms) or a product of sums (maxterms), decoder are used to implement any Boolean function. Let us consider the full adder circuit illustrated. It consists of two outputs, sum (S) and carry (C). The sum and carry are expressed in terms of X, Y, and Z. The Boolean expressions for the outputs sum and carry are as follows:

$$S = x'y'z + x'yz' + xy'z' + xyz$$
$$C = x'yz + xy'z + xyz' + xyz$$

The above expression is obtained by using the OR operation for the combination of output minterms of a 3-to-8 decoder. The output functions of sum and carry are expressed in sum-of-minterms as:

$$S = D_1 + D_2 + D_4 + D_7$$
$$C = D_3 + D_5 + D_6 + D_7$$

The truth table and function clearly shows that the three inputs, X, Y, and Z correspond to eight minterms. Thus, a 3-to-8 decoder is needed to implement this function as is shown in Figure 8.26. The minterms 1, 2, 4, and 7 are applied for sum S and the logical sum of minterms is obtained by connecting a OR gate. Similarly, carry C is implemented by taking logical sum of minterms 3, 5, 6, and 7 from the same decoder.

Table 8.6 Truth table for full adder operation

Decimal value	Input			Output	
	X	Y	Z	S	C
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

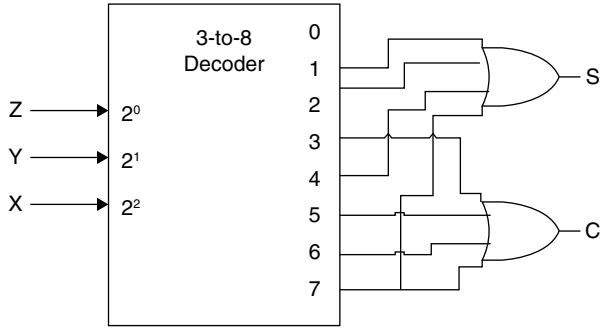


Fig.8.26 3-to-8 decoder circuit

ENCODERS

An encoder is that which operates in the inverse manner of a decoder. It has 2^n inputs, and n output lines. Out of the 2^n inputs only one input can be logic 1 at any given time and this is referred as active input, while all other inputs must be 0. The output corresponding to the active input is generated in binary code. The block diagram is shown in Figure 8.27.

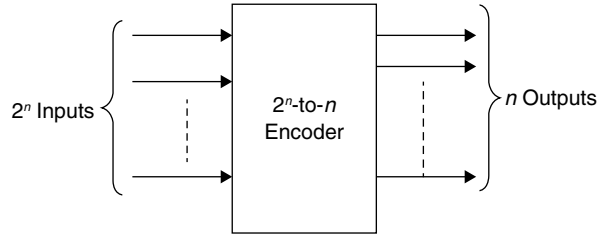


Fig.8.27 Encoder block diagram

To understand the working of an encoder let us consider an octal to binary encoder. A 8-to-3 encoder is shown in Figure 8.28. It consists of eight inputs, one corresponding for each of the octal digits, and three outputs that generate the corresponding binary number.

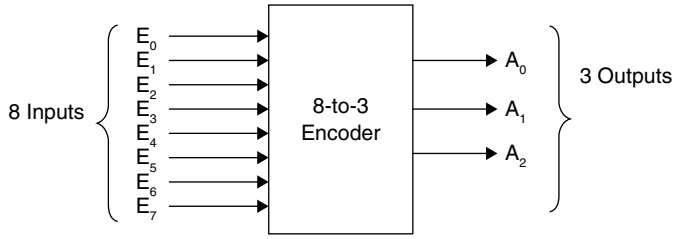


Fig.8.28 8-to-3 encoder

The truth table comprises of eight input variables and three variables corresponding to the output. The output is not obtained for all input combinations. The output is obtained only when one input is equal to logic 1, while all other inputs are logic 0's. As the number of inputs is more (8) implementing a K-map is not easy and hence the output Boolean expression is not obtained.

Table 8.7 Truth table of encoder

Inputs								Outputs			Decimal code
E_7	E_6	E_5	E_4	E_3	E_2	E_1	E_0	A_2	A_1	A_0	
0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0	2
0	0	0	0	1	0	0	0	0	1	1	3
0	0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	0	0	0	0	1	0	1	5
0	1	0	0	0	0	0	0	1	1	0	6
1	0	0	0	0	0	0	0	1	1	1	7

The truth table 8.7 can be used to obtain the expression of the encoder. For octal digits 1, 3, 5, and 7 $A_0 = 1$, therefore,

$$A_0 = E_1 + E_3 + E_5 + E_7$$

Likewise,

$$A_1 = E_2 + E_3 + E_6 + E_7, \text{ and similarly,}$$

$$A_2 = E_4 + E_5 + E_6 + E_7$$

From the above expression it is clear that the encoder can also be obtained by using three 4-input OR gates.

Major Limitation of Encoders

Encoders work only when one of the inputs is active at any given time. When the number active inputs exceeds or is less than 1, then the resulting output will be incorrect.

MULTIPLE CHOICE QUESTIONS

1. A half adder has
   (a) 2 inputs and 2 outputs (b) 2 inputs and 3 outputs
   (c) 3 inputs and 2 outputs (d) 3 inputs and 3 outputs
2. A full adder has
   (a) 2 inputs and 3 outputs (b) 2 inputs and 2 outputs
   (c) 3 inputs and 2 outputs (d) 3 inputs and 3 outputs
3. Half adders are used for addition of
   (a) Three bits (b) Four bits
   (c) Five bits (d) Two bits
4. Full adders are used for addition of
   (a) Three bits (b) Four bits
   (c) Five bits (d) Two bits
5. Ripple carry adders use _____ to add two *n*-bit numbers.
   (a) Two half adders (b) Two full adders
   (c) One half adder and one full adder (d) None of the above
6. An *n*-bit ripple carry adder having '*n*' adders has time delay T equal to
   (a) T (b) *n*T
   (c) 1 (d) 1–T
7. A 1-to-8 demultiplexer has _____ select input lines.
   (a) 2 select input lines (b) 3 select input lines
   (c) 8 select input lines (d) 4 select input lines
8. The circuit containing 2*n* input lines and n output lines, that performs converse of decoding is called as
   (a) Subtractor (b) Decoder
   (c) Multiplexer (d) Encoder
9. In the full adder, if the sum is S and carry is C then
   (a) S = 1 when two or more inputs are unity
   (b) C = 1 when two or more inputs are unity
   (c) S = 1 when all inputs are unity
   (d) C = 1 when all inputs are unity
10. The data from input to a selected output line is directed by
    (a) Multiplexer (b) Coder
    (c) Adders (d) Demultiplexer
11. A 32 to 1 multiplexer has the following terminals
    (a) 32 outputs, one input and 5 control signals
    (b) 32 inputs, one output and 5 control signals

(c) 5 inputs, one control signal and 32 outputs
(d) 5 inputs 32 control signals and one output

12. If a multiplexer can switch one of 16 data inputs to its output then how many inputs does it have?
(a) 16 data inputs and 1 select inputs
(b) 8 data inputs and 2 select inputs
(c) 16 data inputs and 4 select inputs
(d) 2 data inputs and 8 select inputs

13. A one-to-sixteen demultiplexer requires
(a) 1 (b) 2
(c) 3 (d) 4

14. A half adder circuit is constructed using
(a) EX-OR gate and NOR gate (b) EX-OR gate and OR gate
(c) EX-OR gate and AND gate (d) EX-OR gate and NAND gate

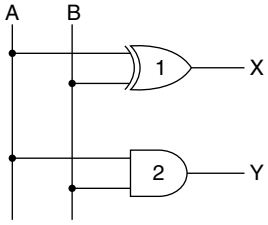
15. Demultiplexer is a device used to change
(a) Sum two parallel data (b) Serial data to parallel data
(c) Parallel data to serial data (d) Sum two serial data

16. A device that coverts binary words into alphanumeric characters is
(a) Decoder (b) Encoder
(c) Multiplexer (d) Demultiplexer

17. To achieve proper functioning of the encoder, which of the following condition is to be satisfied?
(a) All inputs are active (b) Only one of the inputs is active
(c) No input is active (d) All inputs are at logic 0

18. The addition of two binary variables A and B results into a SUM and CARRY equal to
(a) Sum = A'B' + AB, Carry = AB (b) Sum = AB + B, carry = A'B
(c) Sum = AB' + A'B, Carry = AB (d) Sum = AB + A, Carry = A'B'

19. In the half adder circuit given figure if the inputs AB = 10 then



(a) X = 0; Y = 0 (b) X = 0; Y = 1
(c) X = 1; Y = 1 (d) X = 1; Y = 0

Answers

1. (a) 2. (b) 3. (d) 4. (a) 5. (c) 6. (b) 7. (d) 8. (d)
9. (b) 10. (d) 11. (b) 12. (c) 13. (d) 14. (c) 15. (b) 16. (a)
17. (b) 18. (c) 19. (d)

DESCRIPTIVE QUESTIONS

1. Distinguish between combinational logic circuits and sequential logic circuits. Also explain the design requirements of both.
2. Explain the working of a binary half adder with necessary circuit diagram.
3. Draw and explain the binary full adder. Explain the implementation of full adder using a pair of half adders.
4. Explain the construction and working of 4-bit carry look-ahead.
5. Construct a 4-bit adder–subtractor circuit using XOR gates and explain its working.
6. What are binary multipliers? Explain the working of a 2-bit binary multiplier using necessary circuit diagram.
7. What is a decoder? Draw the logic circuit of a 3 to 8 line decoder and explain its working.
8. Construct 4×1 multiplexer and explain its working.
9. Explain how multiplexers are used to implement Boolean functions.
10. What is an encoder? Draw the logic circuit of a decimal to BCD encoder and explain its working.
11. Implement the function $f(A + B + C) = \Sigma (1, 2, 3, 4, 5)$.
12. Explain the working of a 1 to 8 demultiplexer.
13. Write short notes on
    (a) 2×1 multiplexers
    (b) Demultiplexer
    (c) BCD code converter
    (d) Encoder and its limitations

FLIP-FLOPS

Digital circuits are classified into combinational and sequential circuits regardless of their application. In the previous chapters, we have been dealing with logical or combinational circuits. The output of these circuits solely depends on the inputs. Combinational circuits can deliver an output by just applying desirable inputs. These circuits draw their signal paths from left to right. The combinational circuits do not give or store any information related to the previous outputs. A collection of circuits whose output depends on the earlier behaviour of the circuit as well as the present values of the inputs are known as sequential circuits. They work on the feedback principle. Owing to the dependency on past conditions, the sequential circuits are suitable for memory devices. The pictorial representation of the working principles of combinational and sequential circuits is shown in Figure 9.1.

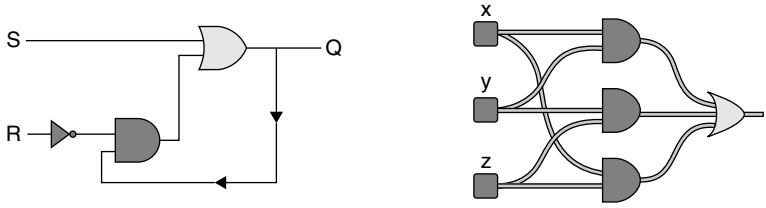


Fig.9.1 Pictorial representation of sequential circuit and combinational circuit

Sequential circuits are also referred as finite state machines (FSMs) which are a more formal name that is often found in technical literature. Since the performance of these circuits depends on the number of states, they are known as finite state machines. Sequential circuits consist of a combinational circuit (present behaviour) to which memory elements (past behaviour) are connected via a feedback path. The general form of which is represented in Figure 9.2.

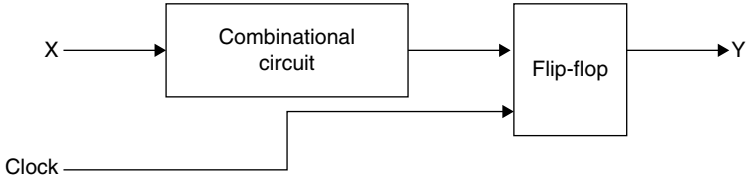


Fig.9.2 General representation of sequential circuit

The sequential circuits are classified into:

- Synchronous sequential circuits
- Asynchronous sequential circuits

Synchronous sequential circuits are those whose operations in most cases are controlled by a clock signal. Circuits that do not consist of a clock signal are known as asynchronous sequential circuits. Synchronous circuits are easier to design and find a wide range of practical applications. There are two ways of constructing sequential circuits namely the Mealy and Moore types. These circuits are constructed when combinational circuits come together with one or more flip-flops as shown in Figures 9.3 and 9.4.

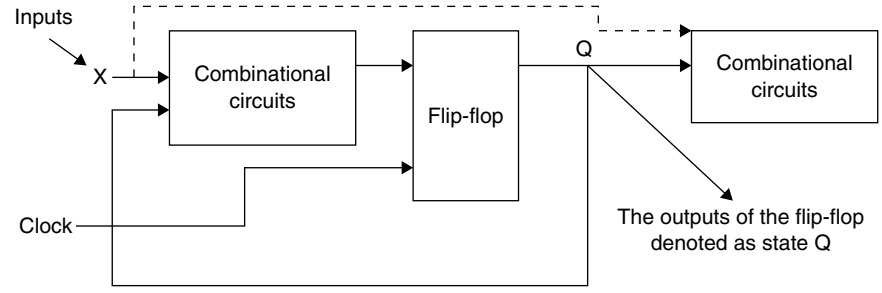


Fig.9.3 Representation of synchronous sequential circuits—Mealy type

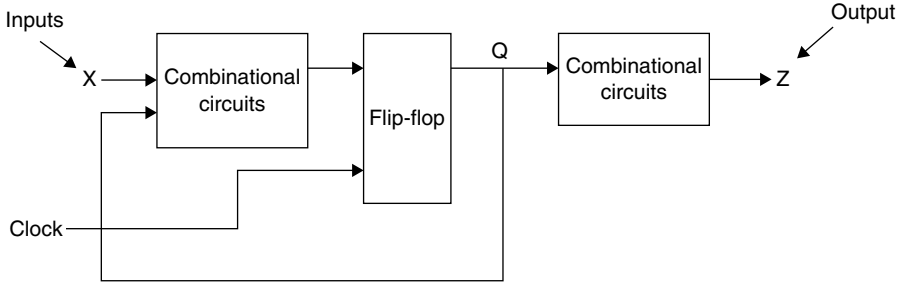


Fig.9.4 Representation of synchronous sequential circuit—Moore type

The output changes of the flip-flop are governed mainly by two things: the clock signal and the input fed by the combinational circuits. This in turn moves the circuit from one state to the next, i.e., $Q_t \rightarrow Q_{t+1}$.

Under the control of the clock signal, outputs of the flip-flop change their state as determined by the combinational circuits that becomes the inputs of the flip-flops. This makes the circuit to go from the current state to the next, i.e., $Q_t \rightarrow Q_{t+1}$. To restrict the flip-flop to single transition for every clock pulse, they must be either positive or negative edge triggered. The combinational circuit that provides the input signals of

the flip-flops derives its inputs from two sources: the primary inputs X and the current outputs of the flip-flops Q. Thus, changes in state depend on both present state and the values of the primary inputs. The outputs of the circuit are generated by another combinatorial circuit, such that the outputs are a function of the present state and the primary inputs. Although the outputs always depend on the present state, they do not have to depend directly on the primary inputs. Thus, the connection represented by dotted line may not exist.

The following steps are to be noted while constructing a sequential circuit:

1. Acquire the requirement of the desired circuit.
2. Derive the states for the machine by first selecting a starting state.
3. Generate a state diagram for this purpose.
4. Create the state table for the state diagram.
5. Apply a procedure to minimize the number of states.

Asynchronous sequential circuits employ the same components as the synchronous but the major difference is that they are not controlled by clock pulses. Asynchronous architecture reduces the power consumption of the systems. They are built towards the emerging concept of clock-less systems. Asynchronous computing systems have demonstrated a full-scaled computational performance to fulfil the budding needs of intensive computational applications. Design and fabrication of asynchronous processing systems is proving to be a promising aspect for future.

Most of the digital circuits are usually an amalgam of combinational and sequential circuits. Despite these differences in operations, the combinational and sequential circuits are simply known as digital circuits as they use the same building blocks—AND, OR, and NOT gates but what makes them different is the pattern they are connected. Thus, the general form of a sequential circuit is shown as an amalgam of combinational circuits along with flip-flops and clocks.

Flip-Flops: "Flip-flop" commonly refers to bi-state devices that are basic memory for sequential logic operations. Flip-flops are used for digital data storage in the form of binary numerical data and their transfer in data banks called "registers". The asynchronous circuits employ core components that have the ability to store information in the form of bits called flip-flops. It can be constructed using NAND or NOR gates that are cross coupled. By increasing the number of flip-flops in a sequential circuit, more information can be stored. The bits of information that are stored define the number of flip-flops to be installed in the sequential circuit. A flip-flop is independent of the time and can hold the information as long as the next input drives it to the switch state. The input of the flip-flop is from both the combinational circuit and the clock. It acts as a feedback loop. These clock pulses alter the state of the flip-flop for each pulse transition. When the clock is stopped, the feedback loop breaks as the flip-flop cannot change its state though the combinational circuit gives its input for the flip-flop to change state. Thus, the transition of the flip-flop from one state to other completely depends on clock pulses.

LATCH

Latches, on the other hand, are level sensitive sequential circuits with two stable states. These are sensitive to the input voltage applied and does not depend on the clock pulse. Flip-flops that do not use clock pulse are referred to as latch. There are two types of latches:

1. SR (or) set–reset latch
2. D (or) delay–direct latch

SR Latch

This type of latch derives its name from set and reset states. They are also known as preset and clear states. A flip-flop and latch are similar, the only difference being the manner in which they are connected. SR latch has two NAND or two NOR gates that are coupled with each other as denoted in Figure 9.5.

The SR latch forms the basic building blocks of all other types of flip-flops. The symbol for an SR latch is shown below in Figure 9.6.

The latch has two inputs S and R and two outputs Q and $\overline{Q}$, Q is the complement of $\overline{Q}$ therefore, when Q = 1 then $\overline{Q}$ = 0 and Q = 0 then $\overline{Q}$ = 1. Q is the normal output and $\overline{Q}$ is the inverted output. When Q = 1 and $\overline{Q}$ = 0, it is called SET or high and if Q = 0 then $\overline{Q}$ = 1 it is called RESET or low. The outputs are cross coupled; the output of one gate is connected to the input of the second gate and vice versa. Thus, each gate is controlled by the other gate. Q_n refers to the previous or preset state and Q_{n+1} the next state. There are different combinations of inputs that S and R terminals take. Since there are two inputs, there are $2^2 = 4$ outputs.

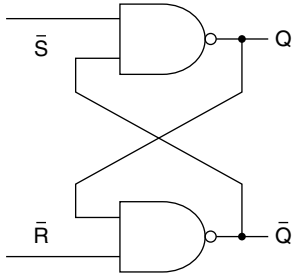


Fig.9.5 SR latch

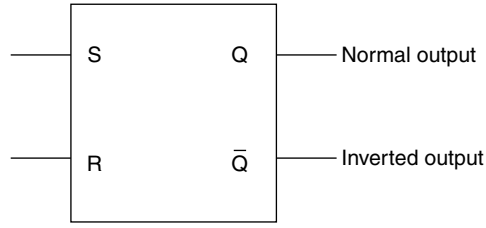


Fig.9.6 Symbol for an SR latch

1. Set (S) = 0, Reset (R) = 0

When the flip-flop is switched ON, one transistor goes to saturation, while the other is cut off giving rise to states of low (0) and high (1) hence initially Q = 1 and $\overline{Q}$ = 0. S = $\overline{R}$ = 0 means the resting state. The output of the first gate is fed to the input of the second gate with R = 0. A NAND gate with inputs 1 and 0 has output 1 and when this output is feedback to gate 1 it gives the output 1 for S = 0.

S	R	Q_n	Q_{n+1}	Q_n	Q_{n+1}
0	0	1	1	0	1

If initially Q = 0 then both outputs become 1 and Q = $\overline{Q}$ at the output this condition is a prohibited state. The latch is in an undefined state as both the outputs are the same.

2. Set (S) = 0, Reset (R) = 1

If we assume $Q_n = 1$ and $\overline{Q}_n = 0$ at second NAND gate then both inputs are 1 such that $\overline{Q}_{n+1} = 0$. Now if both inputs are 0 for gate 1 and output $Q_{n+1} = 1$.

S	R	Q_n	Q_{n+1}	Q_n	Q_{n+1}
0	1	1	1	0	0

If Q = 0 and $\overline{Q}$ = 1, the output of the gate 2 will become 1. This fed to the gate 1 makes the output of gate 1 equal to 1. Now this fed to gate 2 makes output of gate 2 = 0. Hence regardless of the values of outputs of the two gates, the latch is said to be in the SET state.

3. Set (S) = 1, Reset (R) = 0

If we assume $Q_n = 0$ and $\overline{Q}_n = 1$ at second NAND gate then both inputs are 1 such that $Q_{n+1} = 1$ and $\overline{Q}_{n+1} = 0$. Irrespective of the value of Q_n, Q_{n+1} will be 0 and when $Q_n = 0$ the latch is said to be in reset condition.

S	R	Q_n	Q_{n+1}	Q_n	Q_{n+1}
1	0	0	0	1	1

4. Set (S) = 1, Reset (R) = 1

Let us presume, $Q_n = 1$ then both the inputs at gate two is 1 and $\overline{Q}_{n+1} = 0$ then at gate 1 inputs are 1 and 0 so the output is 1.

S	R	Q_n	Q_{n+1}	$\overline{Q}_n$	$\overline{Q}_{n+1}$
1	1	1	1	0	0

It is evident from the above table that the outputs remain unchanged from its previous state.

$$Q = Q_{n+1}; \overline{Q} = \overline{Q}_{n+1}$$

This is called the hold state. Low in both output states is wrong and this is an invalid condition.

SR Latch Using NOR Gates

It is evident from the truth table of NOR gate that its outputs are opposite to NAND gates and hence in this flip-flop the S = R = 0 will have no change or hold state and the S = R = 1 will be the prohibited state. A SR latch using NOR gates is shown in Figure 9.7.

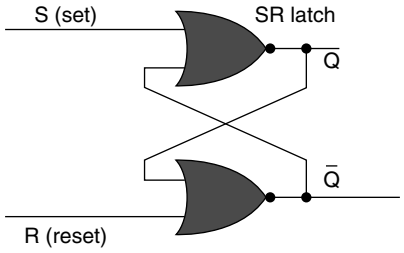


Fig.9.7 A SR latch using NOR gates

S	R	Q_{n+1}	$\overline{Q}_{n+1}$	Mode
0	0	0	0	No change/hold state
0	1	1	0	Set state
1	0	0	1	Reset state
1	1	0	0	Prohibited/undefined state

A NAND SR latch can be converted into NOR SR latch by adding inverter or a single input NAND gate and is shown in Figure 9.8. A NOR SR latch with NAND gates are depicted as follows, and such a latch is called active low SR latch as its inputs are S and R.

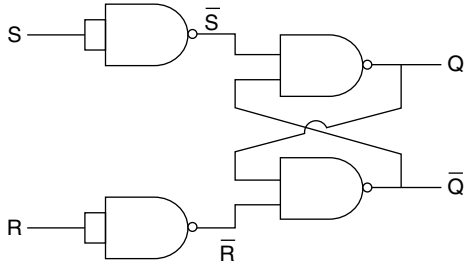


Fig.9.8 SR latch using NAND gates

Thus, SR latch is an asynchronous sequential circuit that solely depends on the propagation delay (t_{pd}) between the NOR gates. Propagation delay is the interval of time required after an input signal has been applied for the resulting output change to occur. The input transition should not be less than 3 t_{pd}.

The SR latch stores a binary bit either 0 or 1 at appropriate conditions and this data is erased once the power is switched off. It is evident from the truth table of NOR latch that Q changes its state from 0 to 1 when S = 1 and R = 0. Similarly, $\overline{Q}$ changes its state from 1 to 0 when S = 0 and R = 1. Therefore, the Boolean expression for set or ON is $Y = S\,\overline{R}\,\overline{Q}$ and the Boolean expression for reset or turn on is $Y = S\,\overline{R}\,Q$. The state diagram can be represented as in Figure 9.9.

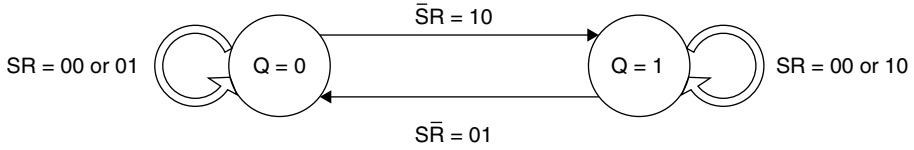


Fig.9.9 State diagram of SR latch

The change of states for different inputs on the basis of the state transition diagram can be represented as:

Q	$\bar{Q}$	S	R
0	0	0	0 or 1
0	1	1	0
1	0	0	1
1	1	0 or 1	0

Contact Bounce Eliminator

Most of the digital circuits are clock driven and have a mechanical switch to fulfil this purpose. When a mechanical switch makes contact to put through a circuit, the switch does not make a solid contact but rather vibrates because of the inherent springiness minutely either making or breaking the circuit. This gives rise to a single pulse that looks more like a waveform and a digital circuit responds to this as several pulses and not as a single clock pulse; therefore to eliminate this, SR latch is interposed between the switch and the system. The SR latch works as a debouncing circuit is shown in Figure 9.10.

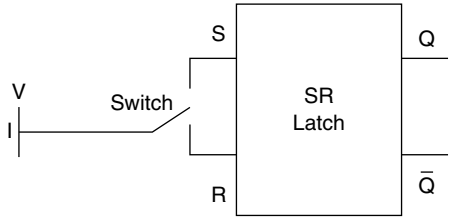


Fig.9.10 Debouncing SR latch circuit

When both S = R = 1, the switch is moved from R to S, then Q changes from 0 to 1 and is latched. Multiple bounces of the switch do not affect the circuit as immediately after the first contact, the latch is set and Q becomes 1. Q can change state only when the switch comes back to R. As long as when S = R= 0 it will remain in the hold state.

CLOCKED SR LATCH

SR latch as discussed is an asynchronous circuit as the output can be changed at any instant of time by changing the inputs, and thus is similar to a combinational circuit. This may result as a limitation when the latch is cascaded in a system and the output

of the latch needs to be altered at some particular instance of time. To overcome this limitation, a clock pulse is introduced that enables to change the state of the latch. This clock is a stable multivibrator, thus the latch becomes a synchronous flip-flop.

The clocked SR flip-flop consists of two NAND gates also known as steering gates with one of the two inputs of NAND gates inputs are labelled as S and R. The other input is inter connected together to the clock terminal. The outputs of the NAND gates are given as inputs to the latch. Thus the outputs are controlled by the clock pulses. The NAND gate equivalent and symbol is shown in Figure 9.11.

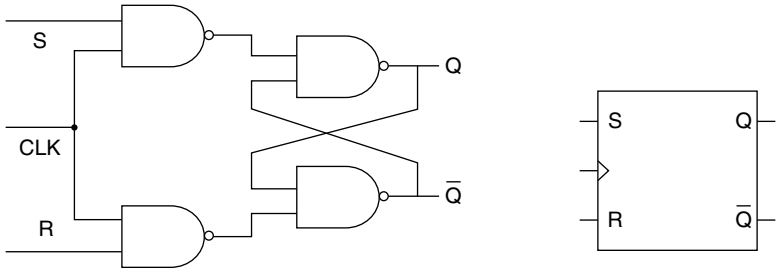


Fig.9.11 Clocked SR using NAND gates

Clock Pulse at Logic 0

When the clock pulse is at logic 0, the NAND gates 3 and 4 have a high output irrespective of the inputs to S and R (either both are 1 or 0).

When S = R = 1 Q and $\overline{Q}$ will be in the hold state.

$$Q_n = Q_{n+1} \quad \text{and} \quad \overline{Q}_n = \overline{Q}_{n+1}$$

The outputs remain in the hold state as long as clock pulse is 0 regardless of S and R flipping between 0 and 1 or S = R = 1.

Clock Pulse at Logic 1 and S = 0, R = 1

When the clock pulse is at logic 1 and S, R takes the above mentioned values then the output A of the NAND gate 3 is at 1 and the output B of NAND gate 4 is at 0. This condition turns OFF the latch and resets it thus,

$$Q_n = 1, Q_{n+1} = 0 \quad \text{and} \quad \overline{Q}_n = 0, \overline{Q}_{n+1} = 1$$

where Q_n and $\overline{Q}_n$ are the states before the clock pulses are applied and Q_{n+1}, Q_{n+1} are states after the clock pulses are applied.

Clock Pulse at Logic 1 and S = 1, R = 0

When the clock pulse is at logic 1 and S = 1, R = 0 then the output A of the NAND gate 3 is at 0 and the output B of NAND gate 4 is at 1. This turns ON the latch and enables it to the set condition

$$Q_n = 0, Q_{n+1} = 1 \quad \text{and} \quad \overline{Q}_n = 1, \overline{Q}_{n+1} = 0$$

Clock Pulse at Logic 1 and S = R = 1

When the clock pulse is at logic 1 and both SR inputs are at HIGH then the output A of the NAND gate 3 and the output B of NAND gate 4 are at 0 thus, disabling the NAND gates. The latch is in a racing condition. This is a prohibited state as the outputs become unpredictable when the clock pulse returns to 0.

The truth table and state table are denoted below. $\overline{Q}_n$, Q_n are states before clock pulse and $\overline{Q}_{n+1}$, Q_{n+1} are states after the clock pulses are applied and 'x' intermediate operation. At the start of the operations, the output Q is maintained at 0 for reset or 1 for set conditions. To achieve this condition, the flip-flop is provided with separate input terminals to gate 1 and gate 2 called preset and clear.

Function/truth table

S_n	R_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Prohibited

State table

S_n	R_n	Q_n	$\overline{Q}_n$	Q_{n+1}	$\overline{Q}_{n+1}$	Mode
0	0	0	1	0	1	Hold state
0	1	0	1	0	1	Reset state
1	0	0	1	1	0	Set state
1	1	0	1	X	X	Prohibited state
0	0	1	0	1	0	Hold state
0	1	1	0	0	1	Reset state
1	0	1	0	1	0	Set state
1	1	1	0	X	X	Prohibited state

CLOCK PULSES

Clock pulses form an integral part of flip-flops. The fundamental abstraction of clock pulses is from the fact that they are used as a regular on/off pulse and the waveform is represented in Figure 9.12. Clock is an eternal analog device and helps in synchronizing operations of different circuit elements. Clock pulses are generally expressed as frequency. Frequency, is inverse of cycle time and expressed in hertz. Frequency of 1 Hz means that there is 1 cycle per second. Hence 1 kilohertz (kHz) means 1000 cycles/sec, 1 megahertz (MHz) means 1 million cycle/sec, 1 gigahertz (GHz) means 1 billion cycle/sec, 1 terahertz (THz) means 1 trillion cycle/sec. But 1 hertz never exist. Maximum clock frequency is defined as the maximum rate that a flip-flop can be reliably triggered.

There are three types of clock inputs:

1. DC or edge triggered
2. AC clocking
3. Master–slave

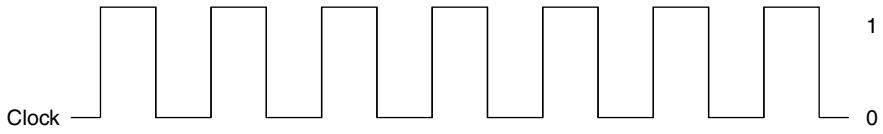


Fig.9.12 Clock pulse representation

DC or Edge Triggered: Consider a clocked SR flip-flop. S and R are the data inputs and CLK denotes the clock input. The SR flip-flop changes state for a HIGH (1) clock pulse. This causes the data to be transported to the output. If the clock pulse has a prolonged duration, then the flip-flop reacts to the next inputs at S and R even before the clock has come back to 0. Thus to evade this, the duration of the clock pulses must be short compared to the pulse repetition rate.

The second problem that arises with clocked flip-flops is the feedback mechanism. As we know sequential circuits are formed when combinational circuits are connected with the flip-flops through a feedback connection. During the sampling of the data, if the flip-flops changes states this causes instability in the circuit. To avoid this, flip-flops are required to toggle only when the clock pulses reach LOW or level 0. This can be achieved by reducing the pulse duration as compared to the propagation delay. The pulse duration can be reduced by introducing edge triggering method. The edge triggering pulses are represented in Figure 9.13(a) and 9.13(b).

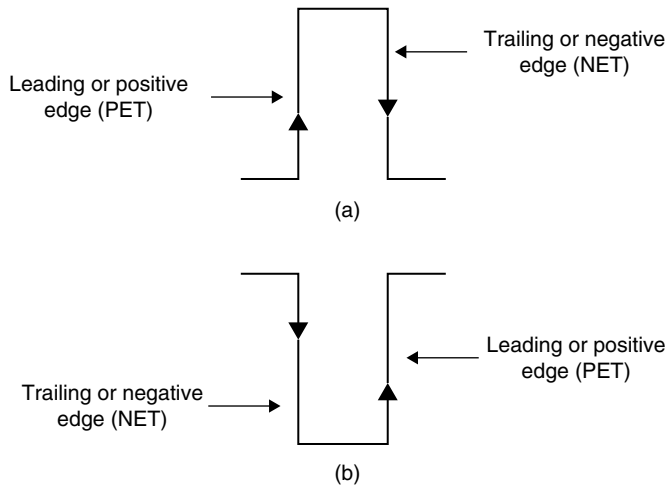


Fig.9.13 **(a)** and **(b)** Edge triggering pulses

Edge triggering is a mechanism where the clock pulses respond only during the positive or negative transition of the clock pulses. A transition of the pulse from 1 to 0 is a positive edge transition (PET) and transition from 0 to 1 is called as negative edge triggering (NET). As spikes have variable transition duration compared to pulses that have fixed duration, the pulses can be converted into spikes and fed to the clock terminals. Spikes can be obtained by feeding the pulses to a differentiator circuit that consists of a capacitor C and resistor R as in Figure 9.14.

During the positive pulse, the capacitor receives a positive impulse that charges it, and during the negative pulse the capacitor receives a negative impulse that discharges it immediately due to the small time constant through the resistor. The spike lasts only for nanoseconds.

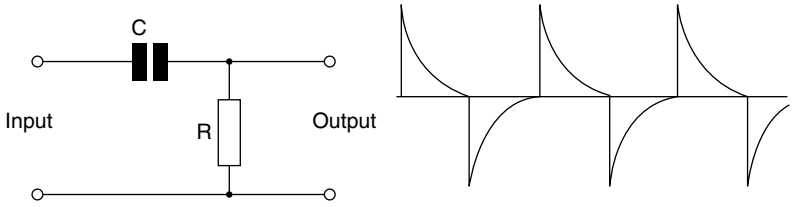


Fig.9.14 Differentiator circuit used to produce spike wave

The symbol for edge triggering SR flip-flop is illustrated as in Figure 9.15.

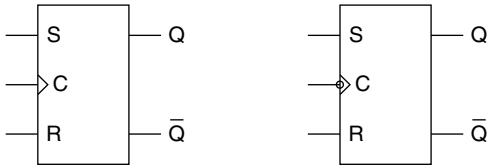


Fig.9.15 Positive edge triggering and negative edge triggering

The data inputs respond to clock spikes only when two timing necessities are satisfied. First, the set up time (t_s) which is the interval required for the logic levels to be maintained constantly on the inputs in this case S and R, prior to the edge triggering of the clock pulse should be minimum for the levels to be reliably clocked into the flip-flop. Second, the hold time (t_H) which is the interval required for the logic levels to remain on the inputs after the triggering edge of the clock pulse should be minimum for the levels to be reliably clocked into the flip-flop. In other words, set up time refers to the time duration for the control inputs to remain in a stable level before triggering and hold time is the interval the control inputs have to stay after the triggering. The set up time duration varies from 10 to 50ns and hold time ranges from 0 to 10ns. The waveform is shown in Figure 9.16.

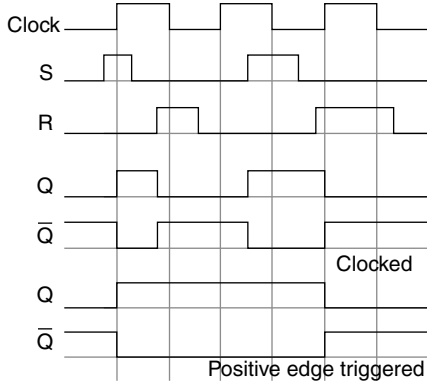


Fig.9.16 The waveform of SR flip-flop

A clock usually generates series of square wave pulses that are positive edge triggered in nature. When the clock pulse reaches 1, both S and R turn LOW (S = R = 0) and sustain in the same state. The output Q is now in HOLD and is 0. Prior to second pulse, S goes to 1 and R is at low but this condition does not affect the previous low state. During the PET of the second pulse and S being high, the clock enables S and R and they toggle thus producing the output as 1. The flip-flop remains in this state for the NET of the second pulse denoted as B though now S becomes 0. For the PET of the third pulse R becomes 1 and S becomes 0 but the output still is 1 because of the positive pulse at S and R inputs. For the next pulse Q is 0 and remains till the subsequent pulse arrives that toggles it to 1. This method is referred to as positive clocking. The flip-flops can be activated through negative clocking also. It is to be noted that when the clock input, S and R, are all given high logic then a race condition is set in the flip-flop which leads to a prohibited state. Hence to avoid this condition, the clock is to be maintained at logic level 0 when S and R are high which is depicted in the timing diagram. The truth table of a clocked SR flip-flop is shown below.

Mode	Inputs			Outputs	
	Clock	S	R	Q_{n+1}	Q_{n+1}
Hold	0	0	0	No change	
	0	1	0	No change	
	0	0	1	No change	
	0	1	1	No change	
Set	1	1	0	1	0
Reset	1	0	1	0	1
Race	1	1	1	1	1

AC Clocking: In this method, the clock is capacitively coupled to the latching mechanism. The clock responds to the transition time and not on the voltage applied. As the method exclusively depends on the clock transitions, the duration of pulses must be less.

Master–Slave Clocking: The master–slave clocking is explained in JK flip-flops.

D LATCH

The letter D in the D latch stands for "data" as this latch stores single bit temporarily. The symbol is shown in Figure 9.17. The D latch transfers stored bit to the output Q when the clock is high or during leading edge and the state of Q remains for clock low. The Q remains in the same state till the next clock high is given to the latch. The state of Q depends and follows the state of D. The working of a D latch resembles a basic memory cell. A D latch is said to be transparent in the clock high as any change in D input reflects at Q output. SR flip-flop can be converted into a D latch by introducing an inverter between both the inputs. S input is reassigned as D input. The usage of a D flip-flop prevails over the undetermined state faced with SR flip-flop.

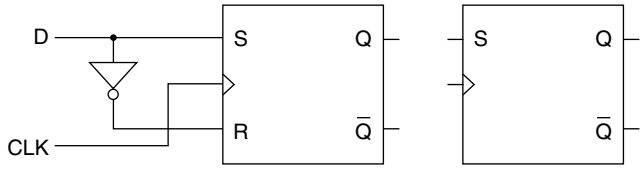


Fig.9.17 D flip-flop sysmbol

In some practical flip-flops, clear and preset inputs are often provided so that it can be desirably tuned to set state (Q = 1) or clear state (Q = 0). These inputs do not depend on the clock signal and are usually asynchronous. D flip-flop with clear and preset terminals is shown in Figure 9.18.

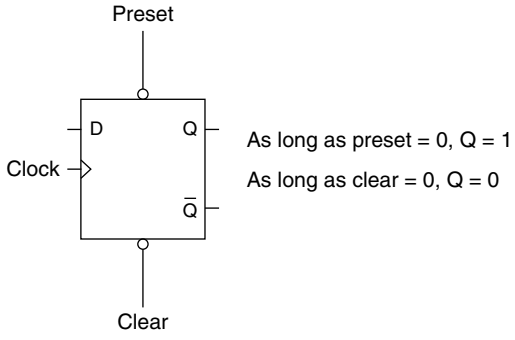


Fig.9.18 D flip-flop with preset terminals

Since the D flip-flop is derived from SR, flip-flop toggles only at two conditions. This condition is when S = D = 1 and R = D = 0 or vice versa. The analysis turns out to be as simple as the transition is only from 1 to 0 or 0 to 1 of the truth table shown. When the clock pulse is 0, the flip-flop is in don't care state as the output can either be 0 or 1. For a high clock pulse, the flip-flop is enabled whose inputs and outputs are as shown.

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

It is noticed from the table that $D = Q_{n+1}$ that is to say that the output follows the input and hence the flip-flop is said to be transparent. Also the output occurs only for alternate clock pulses and thus, a clock delay is set between the input D and output Q therefore, it is also called delay flip-flop.

MASTER–SLAVE D FLIP-FLOP

The master–slave D flip-flop has two individual clocked D flip-flop cascaded together shown in Figure 9.19. First one acts as a master and the second acts as a slave. The master flip-flop dictates the state of the slave flip-flop. When clock is equal to '1'or high, the master flip-flop corresponds to the values of the D input signal but the slave does not change. Thus, output of the master flip-flop (Q_m) follows any changes in D and the output of the slave flip-flop Q_s remains constant. When the clock signal changes to 0, the master flip-flop stops tracking the changes in the D input signal and at the same instance of time, the slave starts responding to the value of Q_m and changes states accordingly.

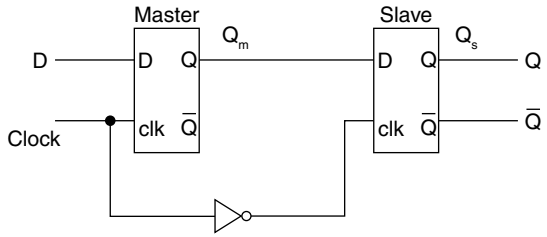


Fig.9.19 Master–Slave flip-flop

But Q_m does not change when clock is 0. Thus, the slave flip-flop undergoes only one change of state during a clock cycle. Thus, the output Q_s of the flip-flop changes for every negative edge or trailing edge of the clock signal. The output waveforms and truth table is represented in Figure 9.20.

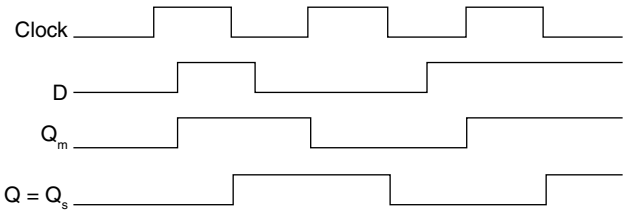


Fig.9.20 Output waveforms of M/S Flip-flop

D	CLK	Q	Q′
1	1	1	1
0	1	0	1
X	0	X	X

EDGE-TRIGGERED D FLIP-FLOP

The master–slave D flip-flop that responded and worked for the positive edge of the clock signal in this type corresponds to negative edge of the clock signal. A negative-edge triggered D flip-flop can be constructed using NAND or NOR gates. The symbol for both positive edge triggered D flip-flop and negative edge triggered D flip-flop is shown in Figure 9.21.

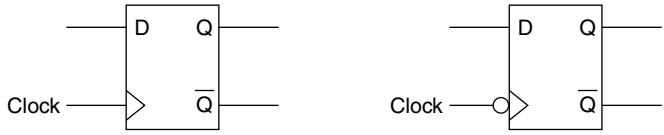


Fig.9.21 Positive edge triggered D flip-flop and negative edge triggered D flip-flop

A circuit, similar in functionality to the master–slave D flip-flop, can be constructed with 6 NAND gates as shown in Figure 9.22. Two latches respond to the D input and clock pulse while the third corresponds to the output for the flip-flop. When the clock is maintained at LOW then S and R inputs are also kept at logic high which causes the output to remain in the same existing state. The data may be 1 or 0. When the clock is 1 then R = 0 and resets the flip-flop making its output, Q = 0. This state is locked and the flip-flop is insensitive to further inputs. If the clock returns to 0, then R goes to 1 which in turn pushes the flip-flop into a dormant state. Now, for D = 1 and clock high, S becomes 0. Therefore, setting the flip-flop that results in output Q = 1 but changes in D while clock is maintained high does not affect the output state.

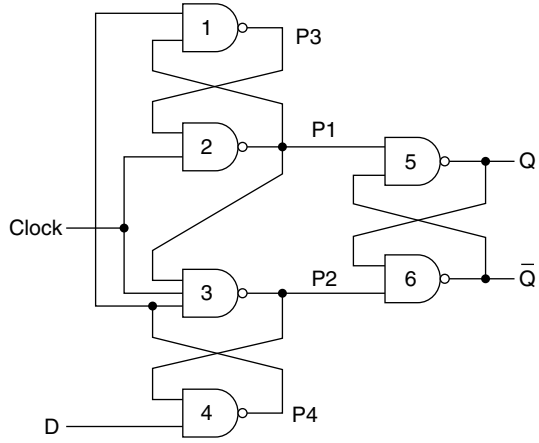


Fig.9.22 M/S flip-flop using NAND gates

Thus, for a positive edge triggered (0 to 1), D flip-flop input D follows output Q but does not respond to a negative transition (1 to 0) of the clock pulse. As this flip-flop also depend only on the clock pulses, the set up and hold time discussed earlier have a vital part in the working of a D flip-flop.

JK FLIP–FLOP

The JK flip-flop is the most versatile of the basic flip-flops. The input of a JK flip-flop has two inputs that are traditionally labelled as J and K with no other significance to JK except being consecutive alphabets. The circuitry of a JK flip-flop is similar to a SR flip-flop with NAND gates and is shown in Figure 9.23. The JK flip-flop can be derived from SR flip-flop by introducing NAND gates 1 and 2 as latch and NAND gates 3 and 4 as steering gates.

If J and K are different then the output Q takes the value of J at the next clock edge. When J and K are both low then no change occurs but when J and K are both high at the clock edge then the output will toggle from one state to the other. This flip-flop performs the actions of SR flip-flop and has an advantage over the ambiguous states. If J and K are interconnected it can also act as a T flip-flop to accomplish toggling action of J and K. This toggle application finds extensive use in binary counters.

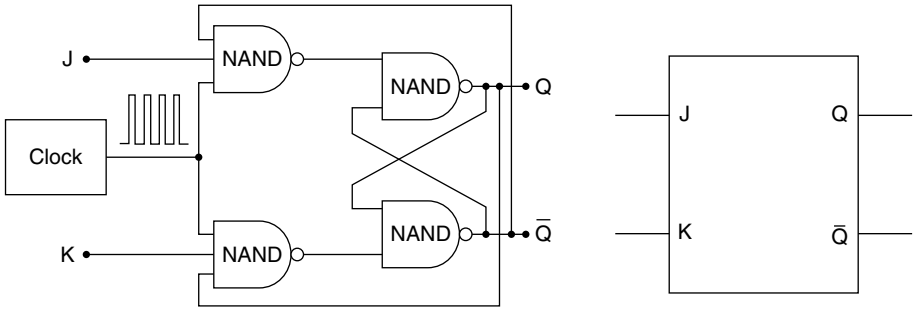


Fig.9.23 Circuit and symbol of JK flip-flop

A simplified version of the versatile JK flip-flop is represented in Figure 9.24. The flip-flop can be constructed using SR flip-flop that receives its feedback from NAND gates. This feedback mechanism is responsible for the toggling action when J = K = 1.

Operation of JK Flip-Flop

1. **When the clock = 0:** Consider a SR flip-flop with AND gates, the output Q is in the hold state, irrespective of S and R values. If a three input NAND gates, 3 and 4, are used at the inputs instead of AND gates, then the output always is 1 for clock = 0. Now if the output Q of gate 1 is same as $\overline{Q}$ then,

$$Q_n = Q_{n+1} \text{ and } \overline{Q}_n = \overline{Q}_{n+1}$$

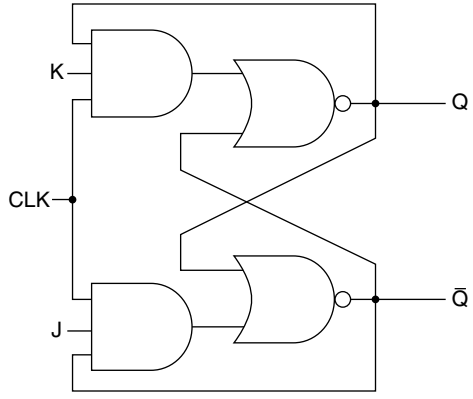


Fig.9.24 JK flip-flop using NAND gates

2. **When clock = 1, J = 0, K = 1:** For $Q_n = 1$, $\overline{Q}_n = 0$ the output of the third gate will be 1 and the output of the fourth gate will be 0. Output of gate 2 which is high is feedback to gate 1 that makes the output of gate 1 logic 0. The state is stabilized at $Q_{n+1} = 0$ and $\overline{Q}_{n+1} = 1$ which is same for SR latch and now the flip-flop resets.

   For $Q_n = 0$, $\overline{Q}_n = 1$, the output of the gate 3 and gate 4 both will be 1. Output of gate 2 which is high is feedback to gate 1 that makes the output of gate 1 logic 0. Therefore,

$$Q_{n+1} = 0 \quad \text{and} \quad \overline{Q}_{n+1} = 1, Q_n = 0$$
$$Q_{n+1} = 0\ Q_n = 1$$

3. **When clock = 1, J = 1, K = 0:** For $Q_n = 1$ and $\overline{Q}_n = 0$, the output of the gate 3 and gate 4 both will be 1. Output of gate 1 will be 1 and the output of gate 2 is 0.

   For $Q_n = 0$ and $\overline{Q}_n = 1$, the output of the gate 3 is 0 and output of gate 4 both will be 1. Output of gate 1 will be 1 and the output of gate 2 is 0. Therefore, the flip-flop goes to set state denoted by the below condition.

$$Q_n = 1, Q_{n+1} = 1$$
$$Q_n = 0, Q_{n+1} = 1$$

4. **When clock = 1, J = 1, K = 0:** For $Q_n = 0$ and $\overline{Q}_n = 1$, the output of the gate 3 is 0 and output of gate 4 will be 1. Output of gate 1 will be 1 and the output of gate 2 is 0. For the next state, i.e., $Q_n = 0$, $Q_{n+1} = 1$, $\overline{Q}_n = 1$, $\overline{Q}_{n+1} = 0$, the output of gate 3 becomes 1 and the output of gate 4 becomes 0. Output of gate 2 will be 1 and this fed to gate 1 makes its output 0. The cycle repeats for the next consecutive state.

   It is evident that the outputs keep changing from 0 – 1 – 0 – 1 – 0 is called toggling which is an exclusive feature of JK flip-flop. By making the SR flip-flop toggle, i.e., S = R = J = K = 1 helps to avoid the prohibited state which makes JK flip-flop versatile. The function Table 9.1 and state Table 9.2 of a JK flip-flop is shown below.

Table 9.1 Function table

Clk	J	K	Q_{n+1}	Mode
1	0	0	Q_n	Hold state
1	0	1	0	Reset, turn OFF
1	1	0	1	Set, turn ON
1	1	1	Q_n	Toggle, ON/OFF
0	0	0	Q_n	No change/hold state
0	0	1	Q_n	No change/hold state
0	1	0	Q_n	No change/hold state
0	1	1	Q_n	No change/hold state

Table 9.2 State table

J	K	Q_n	$\overline{Q}_n$	Q_{n+1}	$\overline{Q}_{n+1}$	Mode
0	0	0	1	0	1	$Q_n = Q_{n+1}$ and $\overline{Q}_n = \overline{Q}_{n+1}$
0	0	1	0	1	0	
0	1	0	1	0	1	$Q_{n+1} = 0$, reset
0	1	1	0	0	1	
1	0	0	1	1	0	$Q_{n+1} = 1$, set
1	0	1	0	1	0	
1	1	0	1	1	0	$Q_{n+1} = \overline{Q}_n$; toggle
1	1	1	0	0	1	

The Boolean expression for the JK flip-flop is given as:

$$Q_{n+1} = J\overline{Q}_n + K\overline{Q}_n$$

And the Karnaugh map of JK flip-flop is given as in Figure 9.25.

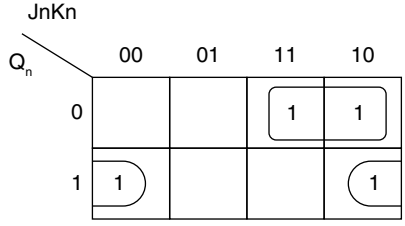


Fig.9.25 Karnaugh map for JK flip-flop

State Diagram: It is manifested from the working of a JK flip-flop that when $J_n = 1$ and clock changes from 0 to 1 then the output Q_n toggles from 0 to 1. When $K_n = 1$ and clock changes from 0 to 1 then the output Q_n toggles from 1 to 0. Hence the state diagram is given as in Figure 9.26.

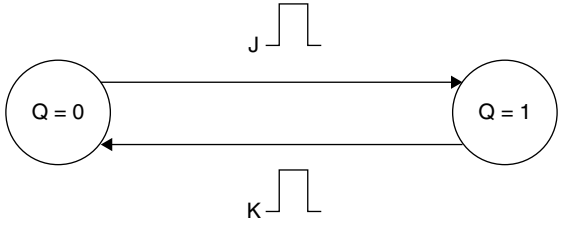


Fig.9.26 State diagram

The JK flip-flop keeps toggling between the 1 and 0 states which is already explained but the toggling is desired only once when JK inputs are high (J = K= 1) and clock = 1. The flip-flop should change from 0 to 1 or 1 to 0 for duration of the pulse but this is not achieved due the following reasons.

Consider a clock pulse of duration t_p and period T. When the data input passes through 2 NAND gates 3 and, it has a propagation delay of $2t_{pd}$. But the duration of the pulse t_p is greater than that of $2t_{pd}$ therefore, this enables the flip-flop to toggle from 1 – 0 – 1 – 0 until the pulse reaches 0. Since the flip-flop keeps toggling continuously it is difficult to predict the output and state is undefined. This is known as racing condition. To overcome this condition, the JK flip-flop can be connected in two different modes.

The Race-around Condition

The SR flip flop does not allow both the inputs to be HICH. This complexity in an SR flip flop is eliminated in a JK flip flop. The inputs of a SR flip flop does not change during a clock pules but this is not true for JK flip flop as it uses the feedback mechanism. When the outputs are given back into the inputs of the gate then this known as feedback mechanism. Consider the JK flip flop whose inputs J = K = 1 and the output of the previous sate Q = 0. A clock pulse of duration t_p and period T is applied. When the data input passes through 2 NAND gates 3 and, it has a propagation delay of $2t_{pd}$. But the duration of the pules t_p is greater than of $2t_{pd}$. After the clock interval that is equivalent to the propagation delay caused by NAND gate connected in series the output will change to Q = 1, which is the complement of the previous state. Now the state is J = K = Q = 1. On applying the next clock pules the output changes again to 0. The output keep changing from 0 to 1 for the duration of the clock pules and at the end of the clock pules the value of Q is uncertain because the value of t is not know exactly. This situation is known as race around condition.

The race around condition can be avoided if

- The propagation delay caused by the NAND gates is greater than the clock interval. But practically this is not possible since the propagation delay in integrated chips is very small.
- A master slave JK flip flop is replaced instead of the JK flip flop. A master slave JK flip flop consists of two SR flip flops connected using feedback mechanism. The output of the second flip flop is connected to the input of the first flip flop.

A positive trailing clock pules is applied to the first flip flop and negative trailing pules is applied to the second flip flop. When the clock pules is at HIGH the output of the first flip flop corresponds to the J and K inputs. During this time the second flip flop is withdrawn since the clock is LOW. When the clock goes LOW, the first flip flop is HIGH. Therefore the outputs of the slave follow the outputs of the master flip flop as shown below. Since the second flip flop simply follows the first one, it is referred to as the slave and the first one as the master. Hence this configuration is referred to as master–slave flip flop. During the clock pules the inputs to the gates in the master flip flop does not change and therefore the race-around condition does not exist. The state of the master–slave flip flop changes at the negative transition (trailing end).

Table 9.3 Truth table of JK flip flop

Inputs					Output
PR	CLR	CLK	J	K	Q
0	0	x	x	x	Race Condition
0	1	x	x	x	1
1	0	x	x	x	0
1	1	x	0	0	No change
1	1	⎍	0	1	0
1	1	⎍	1	0	1
1	1	⎍	1	1	Toggle

MASTER–SLAVE JK FLIP-FLOP

The principle behind the master–slave JK flip-flop is similar to a master–slave D flip-flop. There can be master–slave flip-flops in all three types of flip-flops, SR, JK, and D. A M/S JK flip-flop is obtained by cascading two flip-flops; the first is called the master and second is called the slave. As discussed earlier, the slave follows the master operation. The only difference being the clock signals given to master and clock that determine the output. The clock is connected to the slave through an inverter. The block diagram is shown as in Figure 9.27.

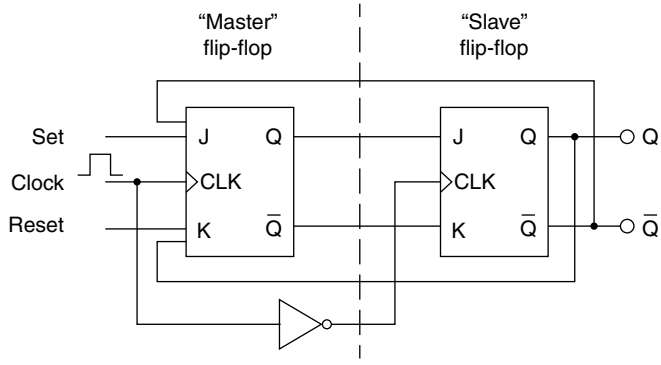


Fig.9.27 Block diagram of master slave JK flip-flop

During the leading edge of the clock pulse, the data inputs are transferred to Q_m and $\overline{Q}_m$ of the master. This state remains as long as the clock pulse is 0 at the slave. When the clock pulse goes negative on trailing edge the master becomes positive at the clock input of the slave and Q_m and $\overline{Q}_m$ of the master flip data inputs are transferred to Q and $\overline{Q}$. This racing condition is averted as feedback from outputs that occurs for pulse at 0. Nevertheless it is to be noted that JK inputs should not be changed for the pulse duration which is overcome in master–slave flip-flop with data lock.

JK Flip-Flop Data Transfer

JK flip-flop has a major application in data transfer. The data is transferred both in synchronous and asynchronous mode. The circuit is represented in Figure 9.28. In synchronous data transfer between two JK flip-flops, a transfer signal on the clock input causes transfer from JK A to JK B. The transfer signal could be applied to several such flip-flops connected in series to create a shift register.

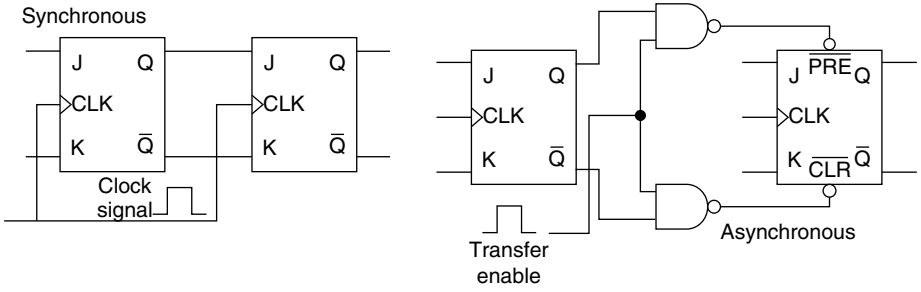


Fig.9.28 Synchronous and asynchronous data transfer using M/S JK flip-flop

In asynchronous data transfer, a transfer pulse may be applied at any time to force the data onto the asychronous set and clearing inputs, storing the data regardless of what is happening on the other inputs.

T FLIP-FLOP

T flip-flop also known as trigger/toggle flip-flop is the fourth type of flip-flop. It can be constructed from SR, D, and JK flip-flops. The block diagram for a T flip-flop is shown in Figure 9.29.

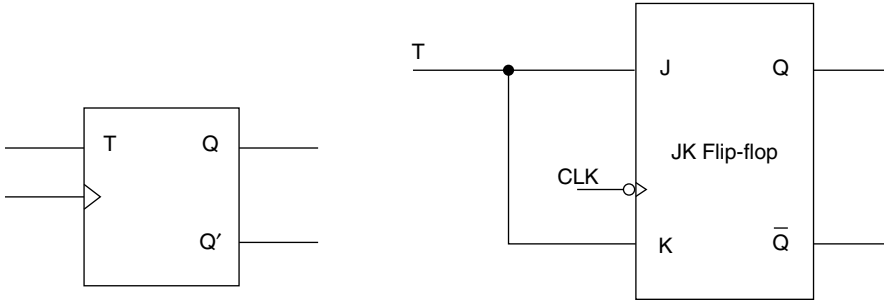


Fig.9.29 Symbol of T flip-flop

The J and K terminals are interconnected. As $J_n = K_n = T_n$, the state table of JK flip-flop reduces to 4 as shown in Table 9.4.

Table 9.4 State table of JK flip-flop

J_n	K_n	Q_{n+1}
0	0	0
0	0	1
1	1	1
1	1	0

The T flip-flop can also be designed as edge triggered and master–slave versions. It is seen from the table that the FF toggles when $T_n = 1$. By keeping $T_n = 1$ permanently the input can be complemented. The waveform for T FF is shown in Figure 9.30.

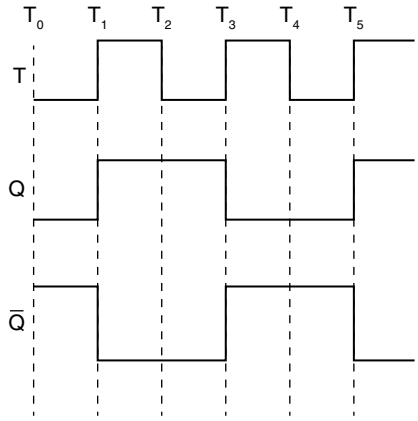


Fig.9.30 Waveform of T flip-flop

CONVERSIONS OF FLIP-FLOPS

The conversion of flip-flops from one type to another is done by connecting a combinational circuit prior to the flip-flop. The output of the combinational circuit is given to the input of the flip-flops. While designing a flip-flop, the excitation tables for both flip-flops are combined and a truth table is made for the data inputs and data outputs. It is then reduced into the K-map and the best logic circuit is deduced. The inter conversion of the flip-flops is discussed below.

SR Flip-Flop to JK Flip-Flop

The conversion of a RS flip-flop into JK flip-flop requires that the inputs of RS flip-flops will be from the combinational circuit and JK flip-flop. As shown in the logic diagram below in Figure 9.31, S and R will be the outputs of the combinational circuit. The truth tables for the flip-flop conversion are given below. The present state is

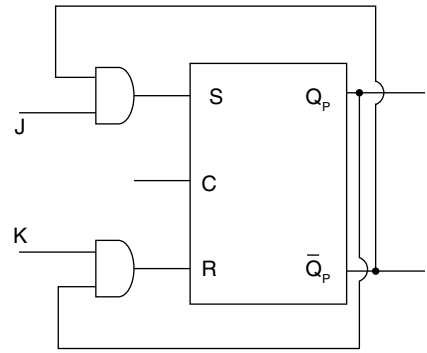
represented by Q_p and Q_{p+1} is the next state to be obtained when the J and K inputs are applied. For two inputs J and K, there will be eight possible combinations. In each combination of J, K, and Q_p, the corresponding Q_{p+1} states are found. Q_{p+1} imply the future states that the JK flip-flop takes. The table is then completed by writing the values of S and R required to get Q_{p+1} from the corresponding Q_p.

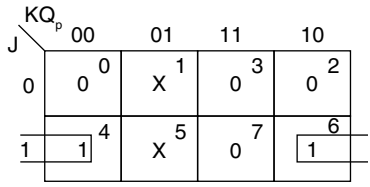
SR flip-flop to JK flip-flop

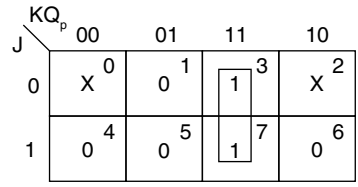
Conversion table

JK inputs		Outputs		SR inputs	
J	K	Q_p	Q_{p+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

Logic diagram







K-Map

Fig.9.31 Truth table, logic circuit and K-map for SR flip-flop to JK flip-flop

JK Flip-Flop to SR Flip-Flop

Here the conversion is obtained by reversing the conversion process where S and R will be the external inputs to J and K. Now J and K will be the outputs of the combinational circuit. Hence the output value of JK flip-flop is defined by S, R and Q_p. Truth table, logic circuit and K-map for JK Flip-flop to SR Flip-flop is shown below in Figure 9.32.

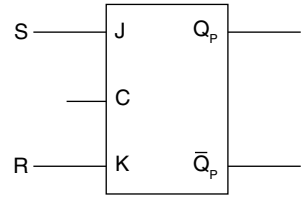
A conversion table is to be written using S, R, Q_p, Q_{p+1}, J, and K. For two inputs, S and R, eight combinations are made. For each combination, the corresponding Q_{p+1} outputs are found. Here both S and R cannot be maintained at logic high which happens to be a forbidden state. Thus, the outputs are considered invalid and the J and K values are taken as "don't care".

JK flip-flop to SR flip-flop

Conversion table

SR inputs		Outputs		JK inputs	
J	K	Q_p	Q_{p+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	X
1	0	1	1	X	0
1	1	Invalid		Do not care	
1	1	Invalid		Do not care	

Logic diagram



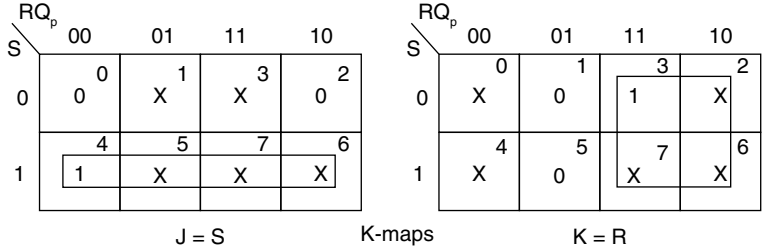


Fig.9.32 Truth table, logic circuit and K-map for JK flip-flop to SR flip-flop

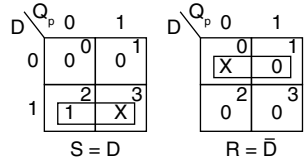
RS to D Flip-Flop: The RS inputs are interconnected. D input is the given to it. D is the external input of the flip-flop and there are four possible combinations. The logic diagram, conversion table, and K-map are shown below Figure 9.33.

SR flip-flop to D flip-flop

Conversion table

D inputs	Outputs		SR inputs	
	Q_p	Q_{p+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

K-maps



Logic diagram

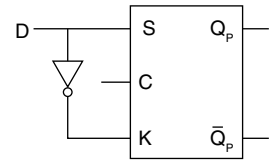


Fig.9.33 Logic diagram, conversion table and K-map for RS to D flip-flop

D Flip-Flop to SR Flip-Flop

Now D is the actual input of the flip-flop and S and R are the external inputs. It has eight possible combinations in the truth table depending on S, R, and Q_p. Again combination of S = 1 and R = 1 are invalid, the values of Q_{p+1} and D are considered as "don't cares". The logic diagram showing the conversion from D to SR, and the K-map for D in terms of S, R, and Q_p are shown in Figure 9.34.

D flip-flop to SR flip-flop

Conversion table

SR inputs		Outputs		D input
S	R	Q_p	Q_{p+1}	
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	1	0
1	1	1	1	0
1	1	1	1	0
1	1	Invalid		Do not care
1	1	Invalid		Do not care

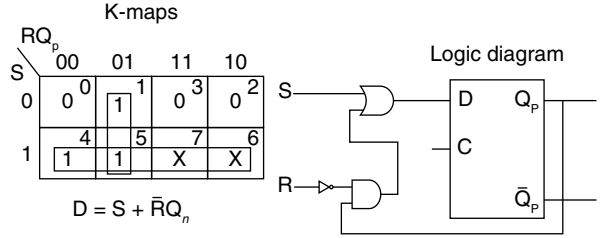


Fig.9.34 Logic diagram, conversion table and K-map for D flip-flop to SR flip-flop

JK Flip-Flop to T Flip-Flop

JK inputs are interconnected and T is the external input for conversion. Four combinations are produced with T and Q_p. J and K are expressed in terms of T and Q_p. The conversion table, K-maps, and the logic diagram are given below in Figure 9.35.

JK flip-flop to T flip-flop

Conversion table

T input	Outputs		JK inputs	
	Q_p	Q_{p+1}	J	K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1

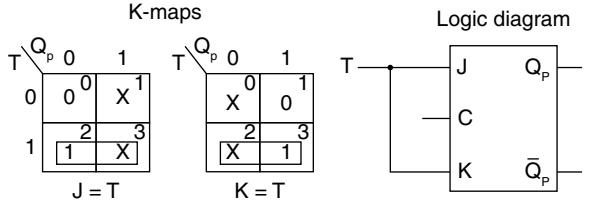
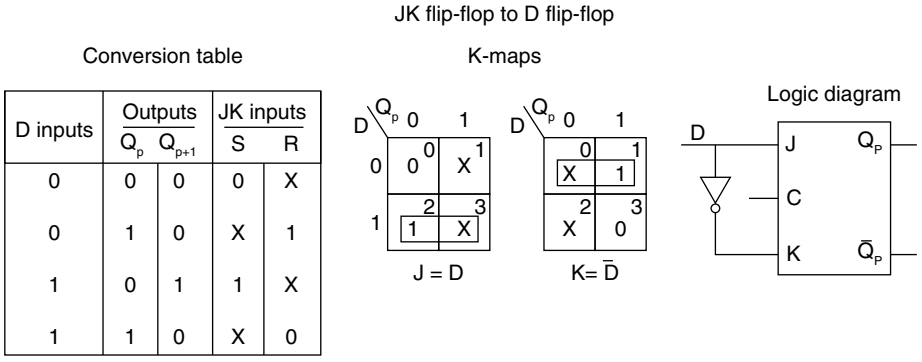


Fig.9.35 Logic diagram, conversion table and K-map for JK flip-flop to T flip-flop

JK Flip-Flop to D Flip-Flop

D is the external input and J and K are the actual inputs of the flip-flop. D and Q_p make four combinations. J and K are expressed in terms of D and Q_p. The four combination conversion table, the K-maps for J and K in terms of D and Q_p, and the logic diagram showing the conversion from JK to D are given below in Figure 9.36.

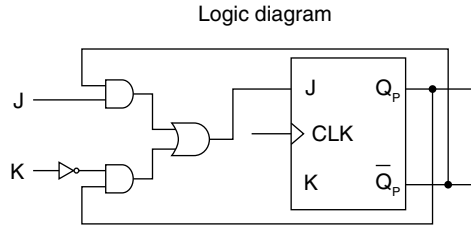


D inputs	Outputs $\bar{Q}_p$	Outputs Q_{p+1}	JK inputs S	JK inputs R
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	0	X	0

Fig.9.36 Logic diagram, conversion table and K-map for JK flip-flop to D flip-flop

D Flip-Flop to JK Flip-Flop

In this conversion, D is the actual input to the flip-flop and J and K are the external inputs. J, K, and Q_p make eight possible combinations, as shown in the conversion table below. D is expressed in terms of J, K, and Q_p. The conversion table, the K-map for D in terms of J, K, and Q_p and the logic diagram showing the conversion from D to JK are given in Figure 9.37.



Conversion table

JK inputs S	JK inputs R	Outputs Q_p	Outputs Q_{p+1}	D input
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

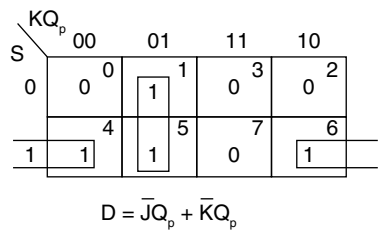


Fig.9.37 Logic diagram, conversion table and K-map for D flip-flop to JK flip-flop

RS to T Flip-Flop

By converting RS flip-flop we can make it to toggle or with a T data input. For the toggling action, the R terminal is connected to the Q terminal and S is connected to the Q' terminal. The logic diagram is shown below in Figure 9.38.

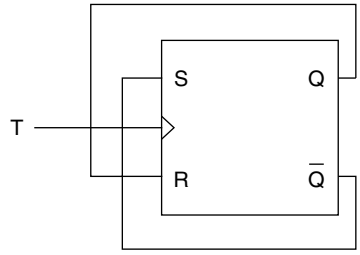


Fig.9.38 Logic diagram for RS flip-flop to T flip-flop

For the data transfer, the external input is by T through the combinational circuit and is given by the R and S terminals as depicted in Figure 9.39.

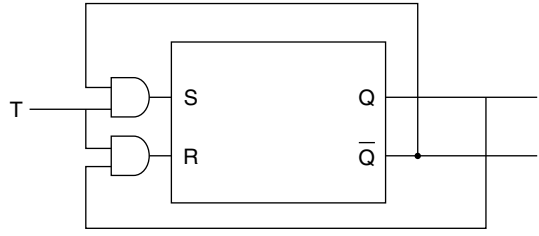


Fig.9.39 Combinational logic diagram for RS flip-flop to T flip-flop

D Flip-Flop to T Flip-Flop

The D terminal is connected to the Q' for converting the D flip-flop for toggling action and for the data input the T input is given to the Q' terminal through the combinational circuit as shown in Figure 9.40.

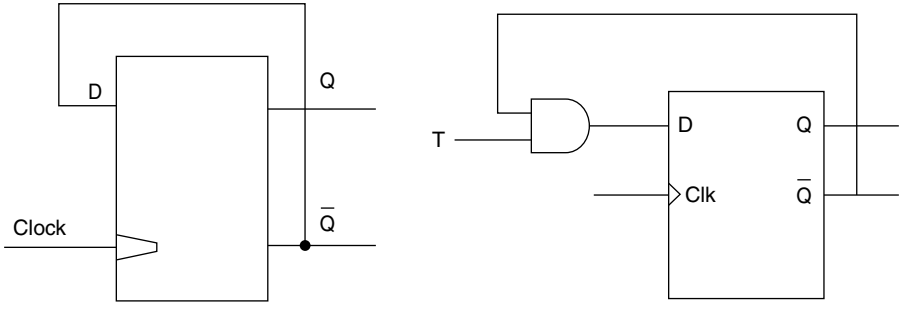


Fig.9.40 Combinational logic diagram for D flip-flop to T flip-flop

APPLICATIONS OF FLIP-FLOPS

Frequency Division

When a pulse waveform is given to the clock input of a T flip-flop formed from JK flip-flop, the Q output is in the form of a square wave. The frequency of the square wave is half of the input clock frequency. The division of the clock frequency depends on the number of flip-flops instituted in the circuit. Further divisions can be achieved by increasing the flip-flops. The circuit and waveform is shown in Figure 9.41.

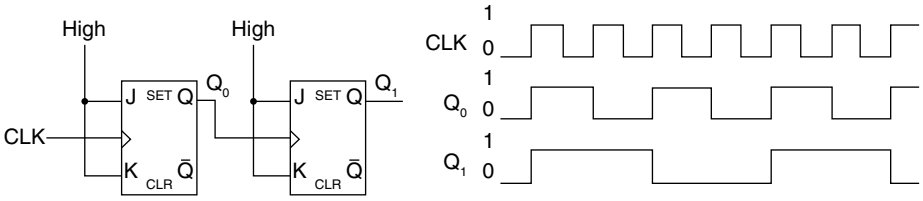


Fig.9.41 The circuit and waveform of frequency divider

The Q output of the second flip-flop is half of the clock frequency and the output of the second flip-flop is one-fourth the frequency of the original clock input. The first flip-flop divides the clock frequency by 2 then is fed to the clock of the second flip-flop which further divided by 2. In general, the division of clock frequency is given as 2^n, where *n* is the number of flip-flops.

Parallel Data Storage

Digital systems store data in groups of bits that represent numbers, codes, or other information. These bits of data available on parallel lines are stored in the flip-flops. This operation is illustrated in Figure 9.42. Each of the three parallel data lines is connected to the D input of a flip-flop. All the clock inputs are connected to a single clock, this makes the data on the D input to get stored simultaneously by the flip-flops for the positive edge of the clock. Registers, a group of flip-flops use for data storage, will be explained in more detail in Chapter 8.

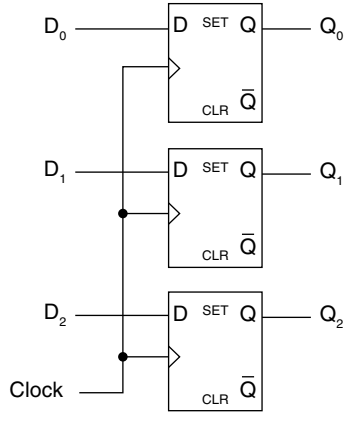


Fig.9.42 Parallel data storage

MULTIVIBRATORS

Multivibrators are two-state devices that find their extensive application in digital electronics. These devices are constructed using resistor coupled amplifiers whose outputs are feedback. Multivibrators are used in storage of binary numbers; to count clock pulses, to perform arithmetic operations but the main application being non sinusoidal wave generation. It generates rectangular, square, saw tooth waves, etc. Multivibrators are of different types:

- Bistable multivibrator
- Monostable multivibrator
- Astable multivibrator

BISTABLE MULTIVIBRATOR

The bistable multivibrators are flip-flops that have two states. It exists in both states for any indefinite time. The state can be changed with external triggering. They are the basic memory devices used in sequential logic. They also function as pulse generator and pulse counter. The active devices in a bistable multivibrator are two n-p-n transistors. Though both the transistors are theoretically identical yet their practical characteristics show differences. This is because a minor difference in the base emitter voltage of one transistor may cause it to conduct earlier than the other transistor. Secondly the positive feedback of the second transistor T_2 makes T_1 to lag behind till it reaches saturation. The voltages of the transistors T_1 and T_2 appear at outputs Q and Q' respectively. Initially let us assume both the transistors to be in OFF state. Now T_1 will be in saturation with zero collector voltage (V_{C1} = 0.2 V to 0.3 V). This is coupled to transistor T_2 and since the voltage is nearly zero T_2 is OFF. The collector voltage of T_2 will be V_{CC} = +5V. The outputs Q and Q' are complementary to each other. If Q is 1, then the output Q' will be 0. The two outputs will remain in this state until the external pulse is applied. When Q is 1 and Q' is 0 then flip-flop is in SET condition. When Q is 0 and Q' is 1 then flip-flop is in RESET condition. Generally all flip-flops are given RESET before the start of the operations. By pressing RESET, a positive pulse is applied to T_2 and the transistor goes to saturation state. This introduces an inversion action on Q that makes it 0.

The state of the flip-flop is changed if a negative pulse is applied at S or when a positive pulse is given at R provided T_2 becomes ON and T_1 is OFF. Generally a negative pulse is preferred over a positive pulse as it is amplified by the use of inverting CE amplifier. Now T_2 is turned ON as the base-emitter voltage is equal to $V_{CC} - V_{BB}$. The transistor that is OFF will have the $+ V_{CC}$ at its collector. This voltage is however not exactly equal to V_{CC} as it is in a voltage divider action when it passes through R_C, R_2 and R_4.

The change of state is achieved by applying a trigger pulse. For triggering a dc pulse is preferred compared to ac since the hold time is more. This causes an increased holding action. Holding action refers to the process where the flip-flop is in ON state as long as the dc is present at the base. An ac pulse can be applied by passing it through a RC circuit coupled with diode as shown in Figure 9.43.

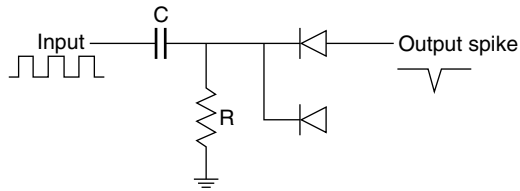


Fig.9.43 RC network used for ac pulse

Consider an ac pulse is connected to the base of the transistors T_1. T_1 is ON and T_2 is OFF. Since the anode is maintained at $+V_{CC}$, the diode conducts for negative pulse and the base emitter voltage of transistor T_1 reduces below the cut off voltage and hence turning it OFF. When another diode is connected to T_2, it is called as a pulse steering network. The introduction of an RC circuit along with diode produces two output cycles one at T_1 and T_2. This is called as RS flip-flop. The circuit and the truth table are shown in Figure 9.44.

R	S	Q	Q′
0	0	0	1
0	1	1	0
1	0	0	1
1	1	X	X

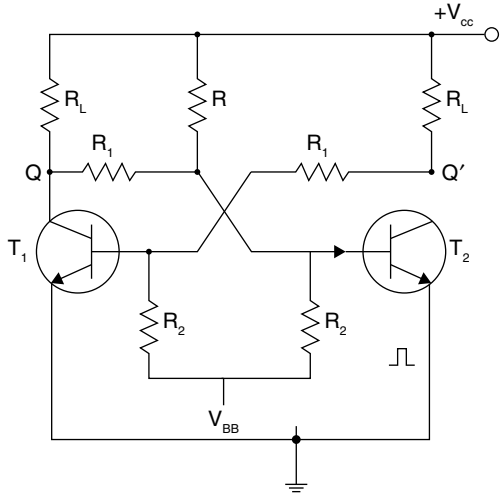


Fig.9.44 Circuit and the truth table of bistable multivibrator

In a bistable multivibrator, the two transistors are turned ON and OFF alternately. This causes a finite delay between ON and OFF operation. This delay or transition time is the time during which conduction transfers from one transistor to the other. This is caused by the stray capacitor C_{BE} that induces an integration effect with the coupling resistors. To reduce the transition delay, the reverse bias voltage at the input

capacitor of the transistor in OFF state must discharge before the base emitter voltage V_{BE} becomes positive. The capacitors connected to increase the transition speed are called as commutating or speed up capacitors.

MONOSTABLE MULTIVIBRATOR

Monostable multivibrators are cross coupled transistors having only one stable state. On applying a trigger pulse it transfers to quasi state. The transition time from the quasi state to the stable state depends on the time constant RC. Monostable multivibrators are also known as One-shot multivibrators as they are used to generate a single output pulse of a specified width on applying a suitable external trigger pulse.

During the stable condition, one of the transistors is ON and the other is OFF. The state can be changed by applying a negative pulse (logic 0) to the ON transistor or a positive pulse (logic 1) to the OFF transistor. The bi-stable multivibrator can be converted into a mono stable multivibrator by replacing the coupling resistor of one transistor by a capacitor and by connecting the base of the same transistor to the main supply. The circuit in Figure 9.45 shows a mono stable multivibrator.

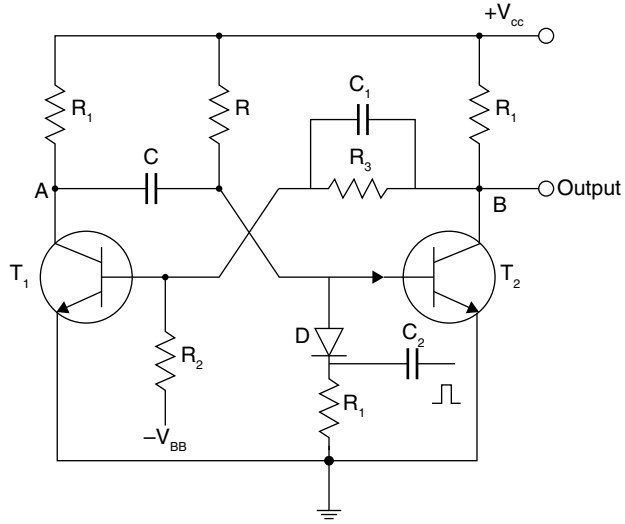


Fig.9.45 Circuit of monostable multivibrator

The collector of second transistor is connected to the base of the first transistor through a parallel RC network formed by R_3 and C_1. And the collector of first transistor is connected to the base of the second transistor through a capacitor C. When T_2 is ON and T_1 is OFF, the multivibrator is in reset state. From the circuit it is seen that the transistor T_2 is also connected to the main supply V_{CC} through resistor R therefore, the base of transistor T_2 is at 0.7 V and its collector is at 0.1 V. Transistor T_1 is also maintained at V_{CC} which charges the capacitor. When the trigger pulse is applied at terminal P, it reaches diode D through R_2, C_2 and produces a negative pulse to the base of the transistor T_2 which pushes the transistor

to the OFF state. The OFF state of transistor T_2 produces its collector voltage to $+V_{CC}$. The collector being connected to the base of transistor T_1 causing a positive potential. This makes T_1 ON. Since T_1 is connected to base of T_2 it goes to a low voltage thereby switching OFF T_2. Thus, T_1 is ON and T_2 is OFF. This condition is known as set state of the multivibrator. The state prolongs till the base of transistor T_2 is increased. This is achieved by the capacitor that gets charged and increases the base voltage of transistor T_2. Now T_2 is driven to ON state and T_1 goes to OFF state. The transition between the states continues for every trigger pulse applied. Waveform of monostable multivibrator is shown in Figure 9.46.

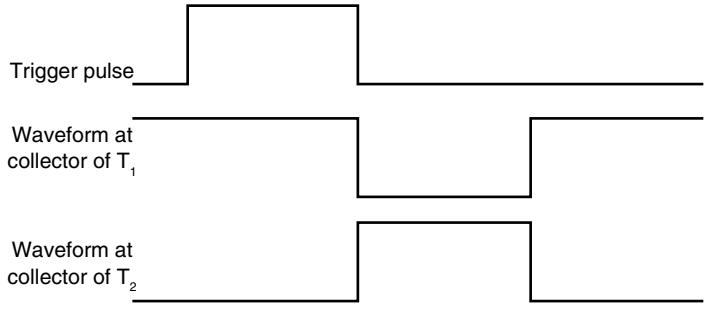


Fig.9.46 Waveform at various terminals of monostable multivibrator

Monostable multivibrators are capable of producing both short pulse and longer rectangular shaped waveform. The leading edge of the waveforms rises with time of the externally applied trigger pulse while its trailing edge depends on RC time constant of the feedback components used. This RC time constant varies with time to produce a series of controlled time delay. The width of the pulse is calculated using the relation.

$$T = 0.693RC$$

On varying the values of R and C the width of the waveform is altered as shown in Figure 9.47.

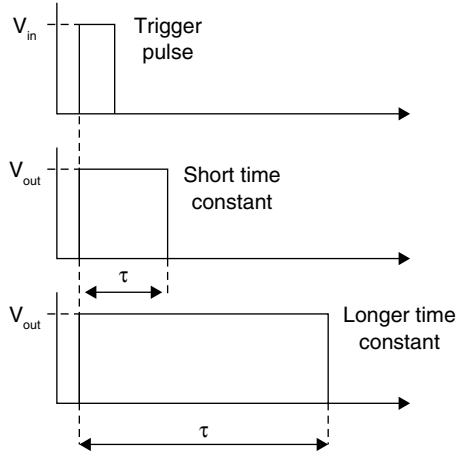


Fig.9.47 Representation of varying width of waveforms

The monostable multivibrators are used in obtaining pulses used in operating a relay from a narrow pulse. It is used to convert random width sized pulses to standard sized pulses. By varying the time constant RC, it can be used as delay circuit.

ASTABLE MULTIVIBRATOR

An astable multivibrator has two cross coupled transistors with both quasi states. The transistors are coupled with each other through capacitors such that they form a feedback network. The capacitors does not allow any of the transistor permanently in ON state. This makes the astable multivibrators to switch continuously between the two unstable states at a constant repetition rate even in the absence of the external trigger pulse. Thus, an astable multivibrator is also known as free running oscillators that can produce continuous square waveform. These are capable of producing pulses when its square wave output is differentiated by passing it through RC network.

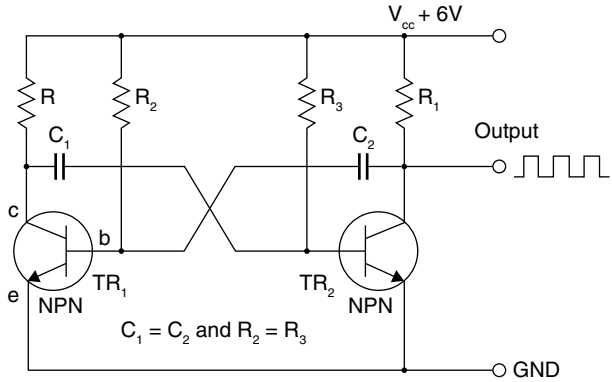


Fig.9.48 Astable multivibrator circuit

The circuit shown in Figure 9.48 has two cross coupled transistors either NPN or PNP. The transistors are coupled through capacitors C_1 and C_2. Since the capacitors are used as feedback components the charge in the capacitors always keeps at least one of the transistors in the ON state. The charge on the capacitors completely controls the multivibrator and the voltage does not change instantaneously.

At a given instant of tine t = 0 let us consider transistor T_1 be OFF and transistor T_2 be ON state. The voltage at the collector of transistor T_1 is $+V_{CC}$ and collector voltage of T_2 is V_{CE}. The base of T_1 has negative potential and base voltage of transistor T_2 is V_{BE}. The capacitor C_2 discharges and this in turn induces an increase in the base voltage of transistor T_1. The voltage rises exponentially till it reaches V_{CC}. Once it reaches the cut off voltage V_γ, transistor T_1 conducts. When T_1 reaches the saturation, the voltage at the collector V_{C1} decreases to V_{CE}, the saturation voltage. Now the charge on capacitor C_1 does not change instantaneously and the fall in the collector voltage of T_1 is applied to base of T_2. Thus, T_2 sets to OFF state. The collector voltage of T_2 increases till it reaches V_{CC} that charges the capacitor C_2. This capacitor is connected

to the base of transistor T_1. This in turn increases the collector voltage of T_2. After the time period t = 0, the base voltage of T_2 rises exponentially and reaches cut off voltage, V_γ. The process is repeated again reverse transition takes place. The waveform are depicted by Figure 9.49.

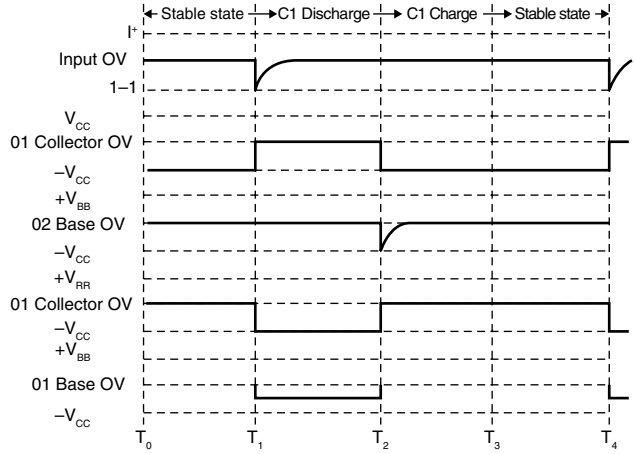


Fig.9.49 Waveform at various terminals

The astable multivibrator is capable of switching between the two states even in the absence of trigger pulse. The time interval t_1 and t_2 are given as

$$t_1 = 0.693 R_1 C_1$$
$$t_2 = 0.693 R_2 C_2$$

The total time period T of the square wave generated is given as

$$T = t_1 + t_1$$
$$= 0.693\ (R_1C_1 + R_2C_2)$$

The frequency of astable multivibrator is given as

$$f = \frac{1}{T} = \frac{1}{0.693(R_1C_1 + R_2C_2)}$$

$$= \frac{1.44}{R_1C_1 + R_2C_2}$$

If $R_1 = R_2$ and $C_1 = C_2$ then

$$T = t_1 + t_2$$
$$= (0.693 + 0.693)RC$$
$$= 1.386\ RC$$
$$f = \frac{1}{T} = \frac{1}{1.386\ RC} = \frac{0.72}{RC}$$

Astable multivibrators can produce two very short square wave output waveforms from each transistor or a much longer rectangular shaped output either symmetrical or non-symmetrical depending upon the time constant of the RC network as shown in Figure 9.50.

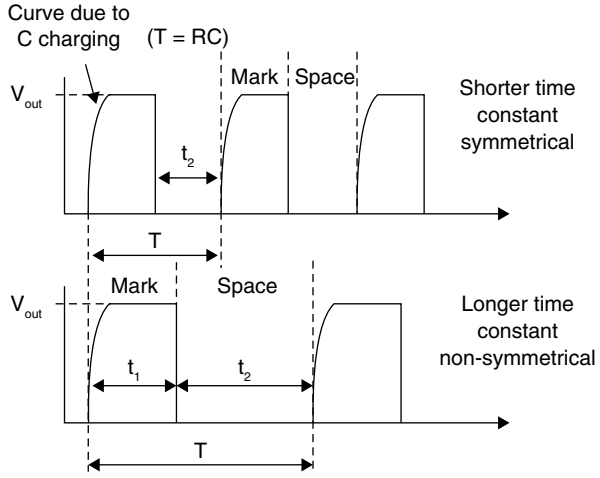


Fig.9.50 Symmetrical or non-symmetrical waveform obtained

SCHMITT TRIGGER

The Schmitt trigger has 2 stable states. The circuit is similar to a bistable multi vibrator the difference being that taken across resistor R_e. The circuit diagram of emitter coupled Schmitt trigger is shown Figure 9.51.

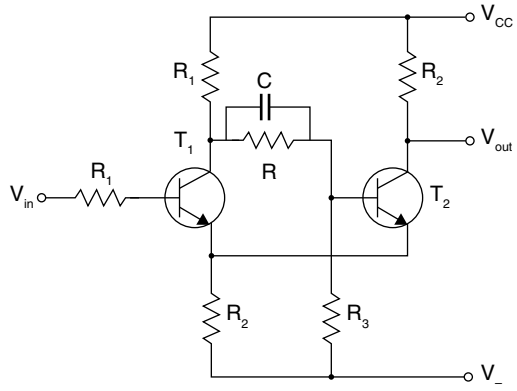


Fig.9.51 Schmitt trigger circuit

The ac voltage applied has namely two regions, upper trip point (UTP) and lower trip point (LTP). When the applied ac input voltage is greater than upper trip point, this

enables the circuit to reach stable state and when the voltage is less than UTP, it causes the circuit to transit to the other state.

When the input signal V_1 is either absent or lower than UTP, it causes T_1 to go to OFF state and T_2 to ON state. This is because, when the transistors ore ON, the current through anyone of the transistors causes a voltage bias across R_E that initiates a reverse bias in T, on the other hand since the voltage passes through voltage divider network made up of resistors R_1, R_2 and R_3, there is a forward bias voltage set in the T_2 which helps it to conduct.

Second case is when the input signal V_1 is greater than VTP, the voltage flow across the emitter base junction of transistor T, this makes the transistor forward biased the base current flown across T and switch it on. Thus, the transistor T, gradually decrease the voltage from the start of conduction. A varying sinusoidal input and its corresponding output are shown in Figure 9.52.

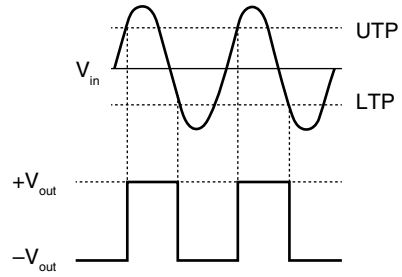


Fig.9.52 Varying sinusoidal input and its corresponding output

Schmitt trigger performs the action of a switching device. It is used in various digital operations and wave shaping circuits. Unlike the outer circuits, the trigger pulse used here is fluctuating ac voltage. This enables the switching operation. There are 2 trip points in the input can be varied gradually without sudden drop. This is known as regenerative switching action. The output keeps changing sharply and can be measured at Q and Q'. The output at Q' is always the complement of output at Q. The feedback mechanism helps in the regenerative action. A apart of the output is feedback to the input which accelerates the action. Care has to be taken to maintain the gain of the feedback loop very large since it initiates the regenerative or feedback mechanism.

IC 555 TIMER

The IC 555 is a simple timer device that finds its applications in various electronic delay circuits. It can perform the function of multivibrators with few modifications. IC is a very simple and cost effective device to use. By connecting resistors and capacitors along with IC 555 we obtain various timing functions. The 555 timer is an 8 pin D-I-L package. The pin out of IC 555 is denoted as in Figure 9.53.

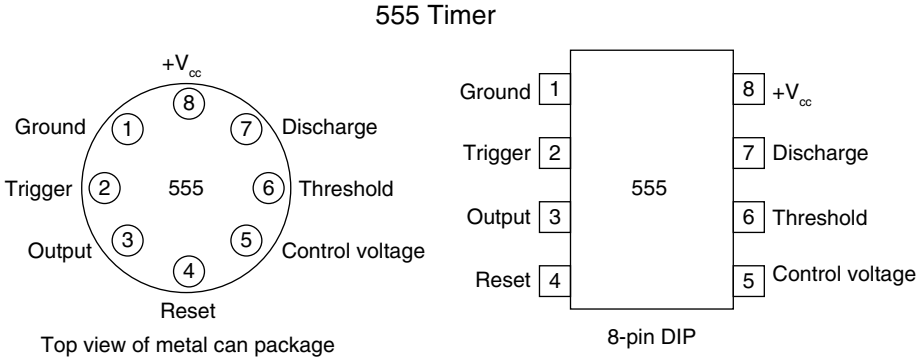


Fig.9.53 Pin out of IC 555

Pin 1—Ground Terminal: This pin is the reference for measuring all the voltages.

Pin 2—Trigger (start) Terminal: This pin provides an inverted input to the comparator. The SET and RESET states of the flip-flop is responsible on the output from this terminal. The amplitude of the external trigger pulse applied to this pin determines the output of the timer.

Pin 3—Output Terminal: The output of the timer is available at this pin. The load is connected across the output terminal in two ways. One, it can be connected to the output terminal either between pin 3 and ground pin (terminal 1). This type of load connection is referred as normally ON load. Second, the load can be connected across pin 3 and supply pin (pin 8) called as normally OFF load.

Pin 4—Reset Terminal: The timer circuit is disabled or reset by applying a negative pulse to this. The device gets RESET and hence it is referred to as reset terminal. A positive voltage ($+V_{CC}$) is applied to the pin when the RESET action is not required. The positive applied to the pin reduces to avoid false triggering. When this pin is not to be used for reset purpose, it should be connected to avoid any possibility of false triggering.

Pin 5—Control Voltage Terminal: The threshold and trigger levels are controlled by this terminal. The width of the output waveform can be adjusted by a giving an external voltage this pin. To avoid any unwanted interference from external signals this pin noise problem, it is connected to ground terminal through a 0.01 micro Farad.

Pin 6—Threshold Terminal: It compares the voltage applied to the terminal with a reference voltage of 2/3 V_{CC}. This is the non-inverting input terminal of comparator. The amplitude of voltage applied to this terminal is responsible for the set state of flip-flop.

Pin 7—Discharge Terminal: This pin has a capacitor connected between this terminal and ground. This pin is named as discharge terminal since the capacitor discharges through the transistor when the transistor becomes saturated. When the transistor is cut-off, the charging of the capacitor depends on the external resistor and capacitor.

Pin 8—Supply Terminal: Positive supply of +5V to + 18V is applied to this terminal with respect to ground (pin 1). The internal configuration of IC 555 is as explained below. It comprises of 25 transistors, 2 diodes and 16 resistors. A functional block diagram of the 555 timer is shown in Figure 9.54.

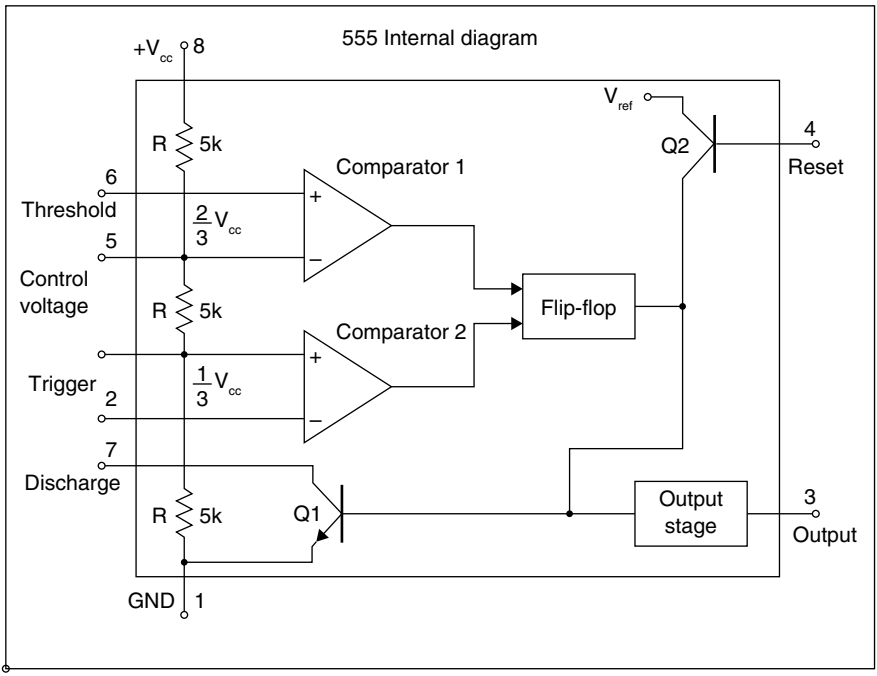


Fig.9.54 Functional block diagram of the 555 timer

It consists of two voltage comparators (C_1 and C_2), an R-S flip-flop, a discharge transistor Q14, a resistive voltage divider (R_3, R_4, R_5), and an output buffer. The negative input of the voltage comparator C_1 is internally connected to the resistive voltage divider. The negative input has a voltage equal to $V_H = 2V_{CC}/3$. This is called the threshold level. The positive input of the comparator C_1 is connected to pin 6 or threshold pin. The positive input of the voltage comparator C_2 is connected to V_1. This voltage is one-third of V_{CC} and is called the trigger level. It is supplied as input to the external trigger pin (pin 2). The input of the voltage comparator C_1 can also be measured at the control pin (pin 5). This voltage is used to adjust the threshold and trigger levels externally. The output of comparator C_1 is the reset input to the flip-flop and the output of comparator C_2 is the set input to the flip-flop. The output of the voltage comparator C_2 goes high when the trigger input falls below the trigger level and hence the flip-flop goes to set state. The output of the voltage comparator C_1 is high when the trigger input is above the trigger level and the threshold input is above the threshold level. This condition resets the flip-flop. The output Q' from the flip-flop drives the discharge transistor Q_D and an inverting output buffer. When the flip-flop output Q' is high, the discharge transistor Q_D is

ON and the voltage at the output pin (pin 3) is low nearing 0. When the output of the flip-flop is low then the discharge transistor Q_D is OFF and the output is high nearing to V_{CC}). The collector of the discharge transistor Q_D is available at the discharge pin (pin 7). The timer operation can be disabled by using the active-low RESET input (pin 4). The output can be ensured to remain 0 irrespective of the output status of the comparators. The dc supply voltage can vary from +5V to +15V and is applied to the V_{CC} (pin 8) and the ground (pin 1). A 5V supply is used for the output, and the RESET input levels as they are compatible with standard TTL or CMOS digital logic circuits.

IC 555 is mainly used in time delay circuits where a delay of 10 microseconds to several seconds can be introduced in the circuit. With the addition of an external capacitor and few external resistors, the 555 can provide the function in the monostable operation, where an output pulse of fixed duration is initiated by a short negative pulse on the TRIGGER input, and in astable operation, where the timer produces periodic output pulses.

MULTIPLE CHOICE QUESTIONS

1. Which of the following conditions toggles the J-K flip-flop?
   (a) J = 0, K = 0 (b) J = 1, K = 0
   (c) J = 0, K = 1 (d) J = 1, K = 1
2. The one-shot multivibrator is also known as
   (a) Bistable (b) Monostable
   (c) 555 Timer (d) Astable
3. When the enable is HIGH for a D latch then the output
   (a) Toggles (b) Is logic 0
   (c) Follows the input at D (d) Complements the input
4. On a master-slave flip-flop, when the gate is HIGH
   (a) Master is enabled (b) Slave is enabled
   (c) Both are disabled (d) Both are active
5. In a flip-flop when the Q output follows the input then the flip-flop is said to be
   (a) Disabled (b) Transparent
   (c) Enable LOW (d) None of the above
6. A J-K flip-flop remains in the previous state when
   (a) J = 1, K = 1 (b) J = 1, K = 0
   (c) J = 0, K = 1 (d) J = 0, K = 0
7. In a positive edge-triggered D flip-flop
   (a) The output toggles when both inputs are HIGH
   (b) Do not care state occurs when both inputs are LOW

(c) The output will follow the input on the leading edge of the clock
(d) Input is transferred to the output on the clock pulse.

8. When the master slave J-K flip-flop is active clock edged then outputs Q and Q' are
   (a) Inactive (b) Complementary to each other
   (c) Q' follows Q (d) Reset
9. In a positive edge-triggered S-R flip-flop, the outputs follows the input when
   (a) Clock is low
   (b) Clock is high
   (c) A low to high clock transition occurs
   (d) A high to low clock transition occurs
10. The HOLD condition of a SR flip-flop is achieved when
   (a) S and R are active (b) S or R are inactive
   (c) S is active (d) R is active
11. The disadvantage of SR flip-flop is
   (a) Presence of do not care state (b) No clock input
   (c) Only one output (d) None of the above
12. Which of the following time is not associated with flip-flops?
   (a) Hold time (b) Propagation delay time
   (c) Interval time (d) Set time
13. When the input of the flip-flop is controlled by the signal transition the flip-flop is
   (a) Toggle triggered (b) Edge triggered
   (c) Pulse triggered (d) Noise triggered
14. The cascading of two JK flip-flops and J-K inputs tied HIGH, the output at the end of fifth pulse is
   (a) 00 (b) 01
   (c) 10 (d) 11
15. The cross coupled NAND gates and NOR gates makes the latch
   (a) Toggle (b) Disabled
   (c) Remain in the latched condition (d) None of the above
16. In a monostable multivibrator the width of the pulse depends on
   (a) Resistors only (b) Capacitor only
   (c) Inductors, resistors and capacitors (d) Resistor and capacitor
17. When J-K flip-flops are cascaded, the output frequency is given as
   (a) Input frequency/2^n (b) Input frequency $\times 2^n$
   (c) Input frequency $\times 2$ (d) Input frequency/2

Answers

1. (d) 2. (b) 3. (c) 4. (a) 5. (b) 6. (d) 7. (c) 8. (b) 9. (c)
10. (b) 11. (a) 12. (c) 13. (b) 14. (a) 15. (c) 16. (d) 17. (a)

DESCRIPTIVE QUESTIONS

1. Elucidate the difference between combinational and sequential circuits.
2. Differentiate the Moore and Mealy type sequential circuits
3. What is a flip-flop? How is it different from a latch? List the applications of flip-flops.
4. Construct a SR latch using NAND gates and explain its working.
5. Construct a SR latch using NOR gates and explain its working.
6. Explain the construction, working and state diagram of the SR flip-flop.
7. Explain a JK flip-flop using NAND gates. Draw the state diagram and Karnaugh map.
8. Draw the block diagram of Master Slave JK flip-flop using NAND gates and explain its working. What is race around condition? How is it eliminated in a master IN slave JK flip-flop?
9. Discuss in detail the conversions of JK flip-flop to other types of flip-flops.
10. Explain the logic diagram for the conversion of T flip-flop to RS and D flip-flops.
11. Explain the working of bistable multivibrator and draw the input and output waveforms
12. Explain the working of monostable multivibrator and draw the input and output waveforms.
13. Explain the working of astable multivibrator and draw the input and output waveforms.
14. Explain the working of Schmitt trigger and draw the input and output waveforms.
15. Describe the functional block diagram and pin out of the IC 555 timer.
16. Write short notes on
    (a) Asynchronous sequential circuits
    (b) SR latch using NOR gates
    (c) Clocked SR latch
    (d) Types of clock inputs
    (e) D Latch
    (f) T flip-flop
    (g) Frequency division using flip-flops
    (h) Contact bounce eliminator
    (i) Parallel data storage using flip-flops

REGISTERS

Registers are binary storage elements that consist of flip-flops and gates. They belong to the family of sequential circuits. The registers are referred as clocked sequential circuits when the combinational circuits are absent. Registers are used to load and store binary information when the clock pulse of the flip-flop circuit is enabled.

The flip-flops in a register can either be JK or D flip-flops. The number of flip-flops used in a register depends on the bits to be stored. If two bits are to be stored then two flip-flops are used. Thus, each flip-flop is capable of storing only single bit. Though the flip-flop stores the data, it is the logic gate that determines the manner in which the data is transferred into the register.

The counter is a special type of register that counts each state but they are different from registers in their working manner. Registers perform the task of data processing such as complimenting, shifting, clearing, incrementing, etc. The process of transferring binary data to a register is termed as loading.

Registers find their applications in a wide range of devices. They form the temporary memory devices in calculators, sequence generators, and generate sequenced frequency pulses. They are applied in time delay circuits, parallel to serial conversion in transmission line, as special counters. Registers are used in designing microprocessors that form the brain of computers. They are seven 8-bit general purpose register and five 1-bit register known as flags.

SHIFT REGISTERS

If a register apart from storing bits of information also shifts the stored bits from one register to another it is called as shift register. Each flip-flop of shift register is a D type. SR or JK can be converted into D flip-flops. D flip-flop is chosen as its output is equal to input when it is clocked. An SR or JK flip-flop is converted into a D flip-flop by giving an inverted input of S to R and J to K. All flip-flops are clocked simultaneously so that the circuit is a synchronous sequential logic circuit. The input is given to the first flip-flop only. The output of the flip-flop Q and Q are connected to J and K or S and R of the successive flip-flop. This is known as double rail output. In a single rail output, the Q terminal is connected to input D of the succeeding flip-flop. If the input at the first flip-flop is 1 and edge triggered by a clock pulse, the output of the first flip-flop is 1.

The bit stored in the flip-flop is shifted by 1 stage and the bit stored at the last flip-flop is last. Each shift register is referred to as a stage.

In a double rail output, an inverter at R terminal of SR flip-flop or K terminal of JK flip-flop can be avoided as output Q is taken from previous flip-flops. Also another reason to eliminate the inverter being that the clock triggers only for positive edge.

Classifications of Shift Registers

Shift registers are classified under four types, namely:

1. **Serial In/Serial Out (SISO):** One bit of data can be moved in and out of the register serially at a time.
2. **Serial In/Parallel Out (SIPO):** One bit of data can be loaded serially but the stored data can be taken out of the register parallely.
3. **Parallel In/Serial Out (PISO):** The data can be loaded simultaneously but can be removed only as one bit at a time for each clock pulse.
4. **Parallel In/Parallel Out (PIPO):** The data can be loaded and read out into the register simultaneously. There is also a bidirectional shift register of this type where the inputs can be shifted right to left or left to right.

The schematic representation of the types of registers are shown in Figure 10.1:

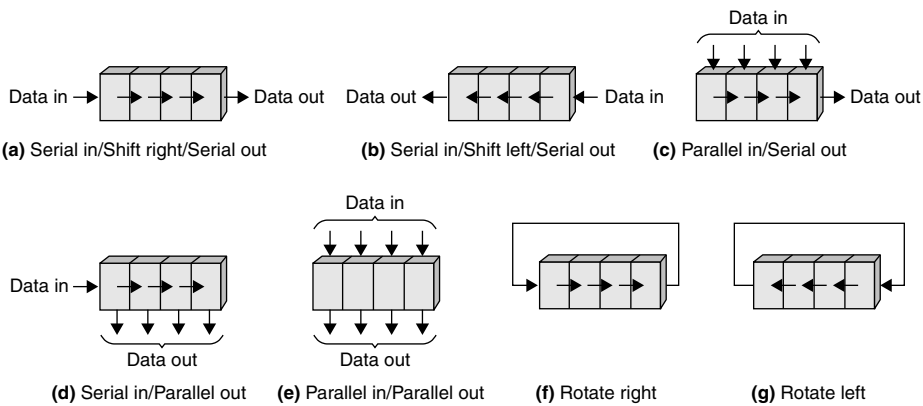


Fig.10.1 Basic data movement in shift registers

Serial In/Serial Out (SISO): This type of shift register loads the data and shifts it from left to right if it is unidirectional and in both directions if it is bidirectional register. The least significant bit of the binary number is stored in the left most flip-flop and keeps shifting to get the final output at the right most flip-flop.

A 4-stage shift register using JK flip-flop is shown in Figure 10.2. The bit is loaded at the flip-flop A and output is obtained at flip-flop D after a delay of 4 clock pulses. The NAND gate connected at the input performs the gating function. The input at A is passed to the flip-flops when B = 0 and is inhibited when B = 1.

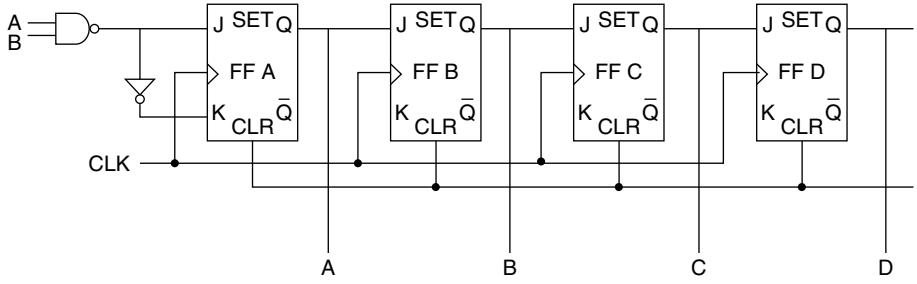


Fig.10.2 4-stage shift register

The operation of a 4-stage register shifting data 1010 from left to right is shown in Figure 10.3.

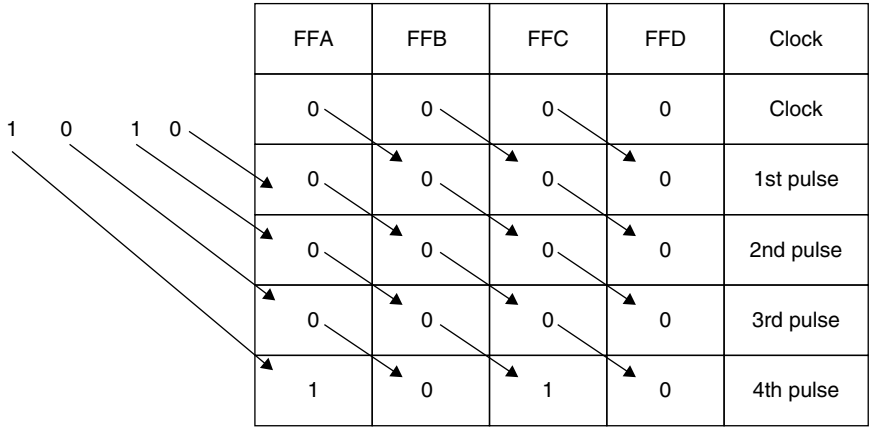


Fig.10.3 Operation of 4-stage shift register

The least significant bit of the binary number is stored on the left most flip-flop A. This data is shifted to the next flip-flop B on the right at the first clock pulse. For the second clock pulse, the information stored in FFB is shifted to FFC. At the end of the third clock pulse, the bit stored in FFC is shifted to FFD. Now the data is finally taken at the output D of the right most flip-flop D. Thus, the flip-flop takes the state of the previous flip-flop for each trailing edge of the clock pulse. SISO register shifts the data serially with a rate of one bit per pulse. The complete number is stored in the register at the

end of the fourth pulse. Now to get the entire data out of the register, four more pulses are required. For the fifth pulse, the FFA is cleared and one bit of the stored information is lost. At the completion of the sixth pulse, two bits of the stored data are lost and this continues till all the data is lost which is achieved at the completion of the eight pulse. Figure 10.4 illustrates the loading of the four bits (1010) serially into the register.

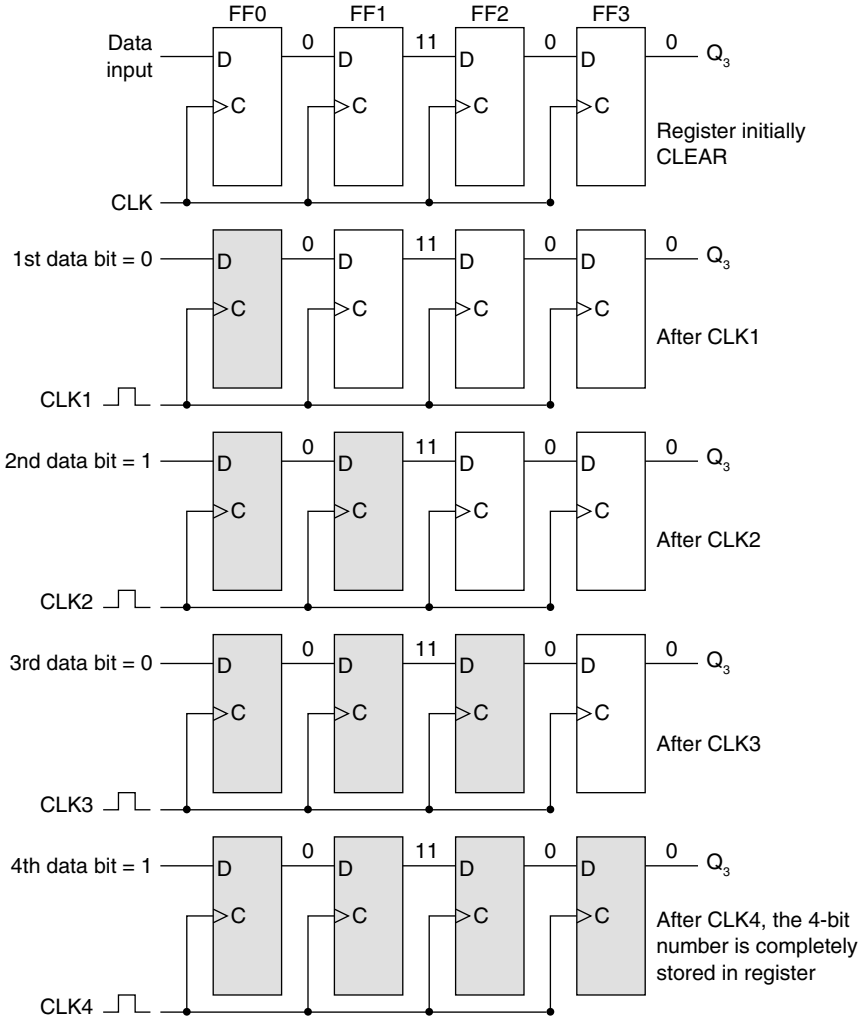


Fig.10.4 Data (1010) loaded into SISO register

Figure 10.5 shows the four bits (1010) being serially shifted out of the register and replaced by all zeros.

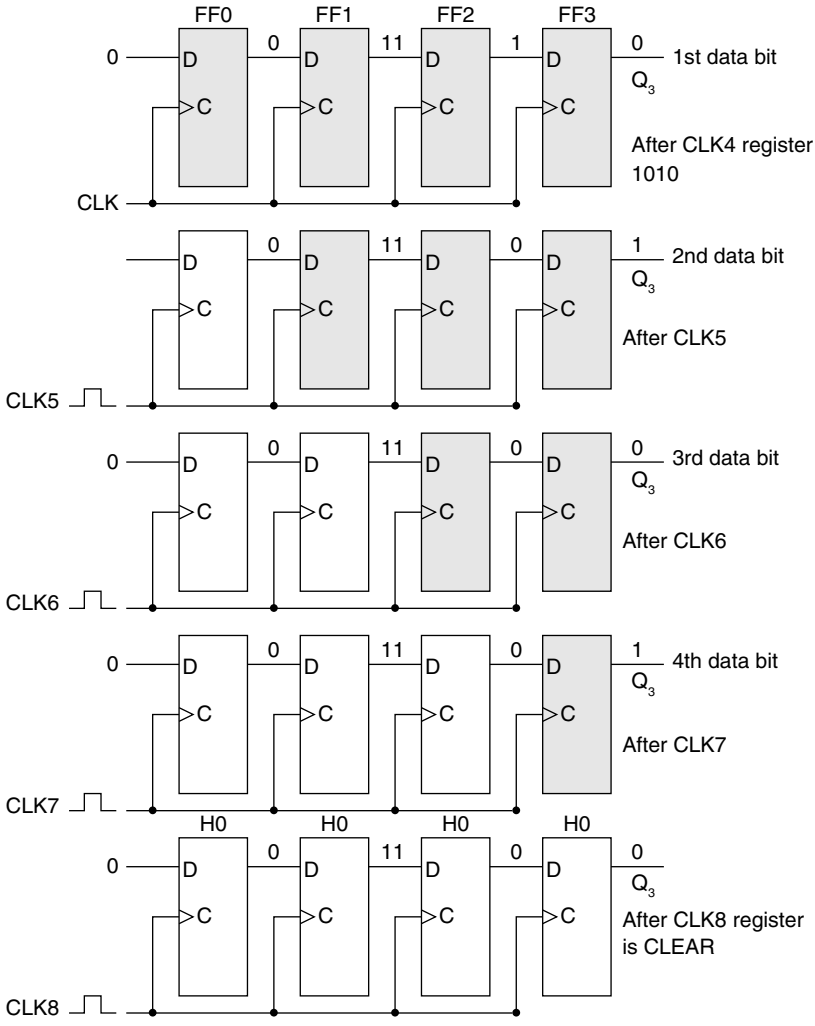


Fig.10.5 Data 1010 shifted out of SISO register

A series of bits are shifted in and out of the register at the same time. A traditional logic block symbol for an 8-bit serial in/serial out shift register is shown in Figure 10.6. The "SRG 8" designation refers to a shift register (SRG) with an 8-bit capacity.

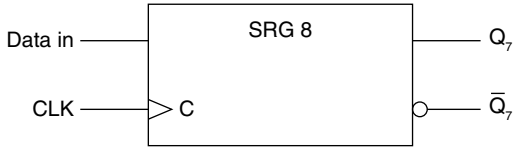


Fig.10.6 Block symbol of 8-bit SISO

Serial In/Parallel Out (SIPO) Shift Register: The circuit of a serial in/parallel out register using D flip-flop and block symbol of 4-bit SIPO is shown in Figure 10.7(a) and (b) respectively. It is similar to a serial in serial out (SISO) register the only difference being the manner by which the output is taken out of the register. The data is entered serially one bit/pulse and all the 4-bit data is loaded into the register. Now the output can be read simultaneously through AND gate that helps in storing the data to be taken out of the register.

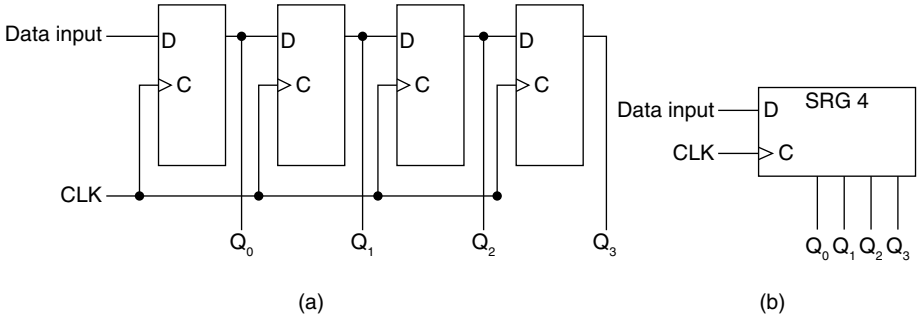


Fig.10.7 (a) Circuit of SIPO (b) 4-bit SIPO-block symbol

The waveform of the serial in parallel out 4-bit register is shown in Figure 10.8. The serial in parallel out register also requires 4 pulses to load the input data, i.e., one pulse for each bit.

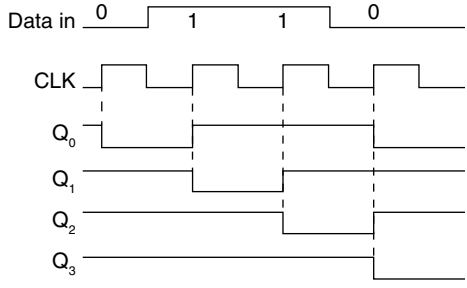


Fig.10.8 Waveform of 4-bit SIPO

The register requires only one clock pulse to read the data at the output. IC 74LS164 is an 8-bit register. The logic diagram and a typical logic block symbol are shown in Figure 10.9. It has 2 gated inputs A, B and a clear terminal which is active low. The output terminals denoted as pins 3 to 7 and 10 to 13 help in obtaining the parallel output. The serial output can be read at pin 13.

The sample timing diagram for the 74LS164 is shown in Figure 10.10. The serial input data at A is shifted into register when the input B goes HIGH.

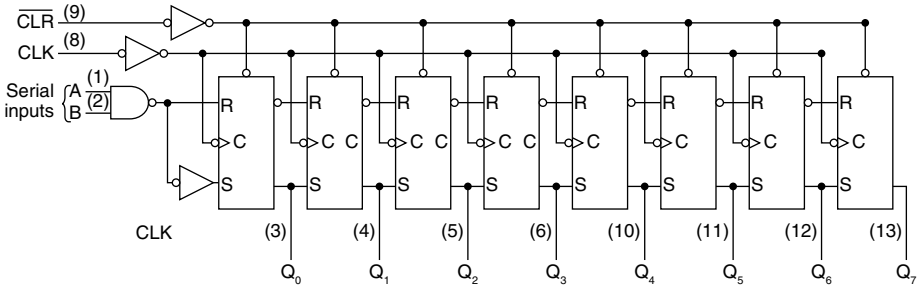


Fig.10.9(a) Logic diagram of 8-bit SIPO

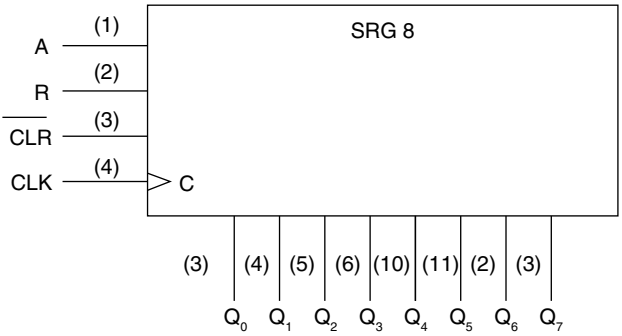


Fig.10.9(b) Logic symbol of 8-bit SIPO

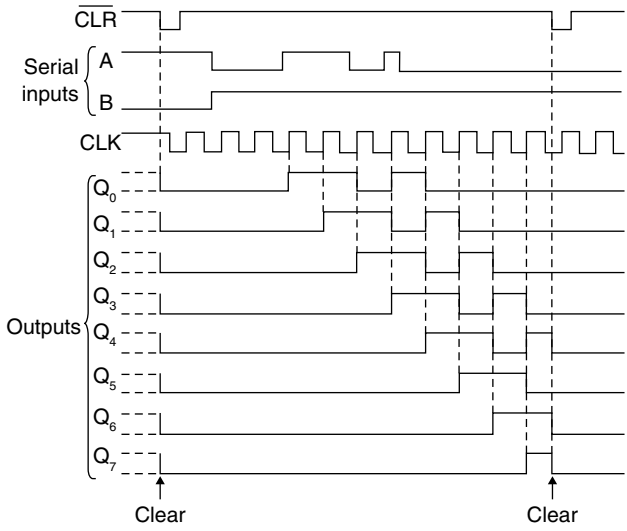


Fig.10.10 Timing diagram of 8-bit SIPO

Parallel In/Serial Out (PISO) Shift Register: A parallel in/parallel out register has parallel data inputs. The bits are entered simultaneously into their particular stages through parallel lines instead of a bit-by-bit entry as in the case with serial data inputs. The serial output is taken out in the same way as described before. This register can be constructed either using DFF or SR flip-flop. A 2-bit (PISO) register using D flip-flop is shown in Figure 10.11.

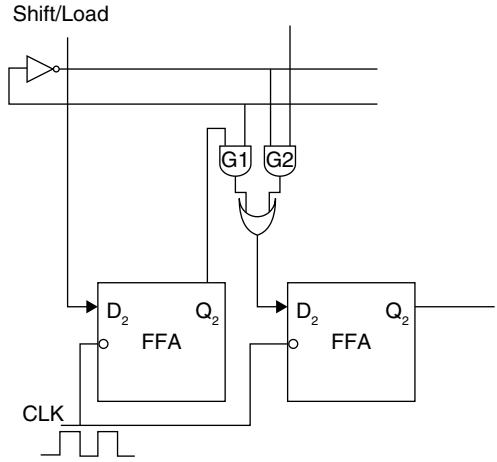


Fig.10.11 2-bit PISO register

The number of flip-flops keeps increasing with the number of bits to be added in a parallel manner. When the load/shift is low, AND gate G_1 is enabled and G_2 is disabled. So that the data at A, B is available at D input of respective flip-flops. When a clock pulse is applied, the inputs are transferred to the Q outputs. When load/shift is high, G_1 is disabled and gate G_2 is enabled. Thus, that output $Q_A Q_B$ at respective stages is shifted to the successive flip-flops. The second type of PISO constructed using SR flip-flop is shown in Figure 10.12.

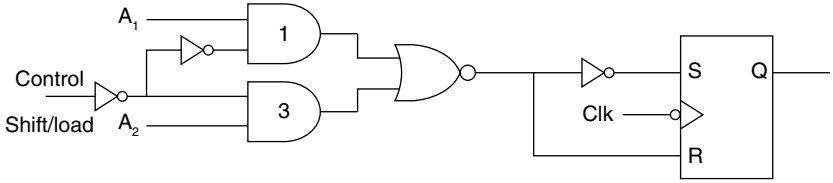


Fig.10.12 PISO constructed using SR flip-flop

Control Low: When the control load/shift is low, AND gate 2 is enabled and gate 1 is disabled. If the A_2=0, then the output at gate 2 is 0. Now the one of the input to gate 1 is 0, the output will also be 0 irrespective of the other input being 1 or 0. This makes the output of the NOR gate high. This output is clocked and given to S making it low and clocking the bit at S is transferred to Q. If $A_2 = 1$, then output of gate 2 is 1 and that of gate 1 is 0. NOR output is low, is clocked and fed to SR flip-flop which makes input to S high thereby making Q = 1.

Control High: A high at the control and $A_2 = 1$, $A_1 = 1$ makes the output of NOR gate low. Input S is high. Thus, the output follows S.

For $A_2 = 0, A_1 = 0$, the output of NOR is high. Thus, clocking input data is available at S. The bits involved determine the shift registers in the centre. The number of registers can be increased in the circuit by connecting Q terminal to A_1 of the succeeding stage.

Consider a 4-bit register with the parallel input data. The clock and SHIFT/LOAD waveforms are given below. The logic diagram is depicted in Figure 10.13. When the first clock pulse is given, the parallel data ($D_0D_1D_2D_3$ = 1001) are loaded into the register, making Q_3 as 1. On clock pulse 2, the 0 from Q_2 is shifted onto Q_3; on clock pulse 3, the 0 is shifted onto Q_3; on clock pulse 4, the last data bit (1) is shifted onto Q_3; and on clock pulse 5, all data bits have been shifted out of the register (assuming the D input remains a 1).

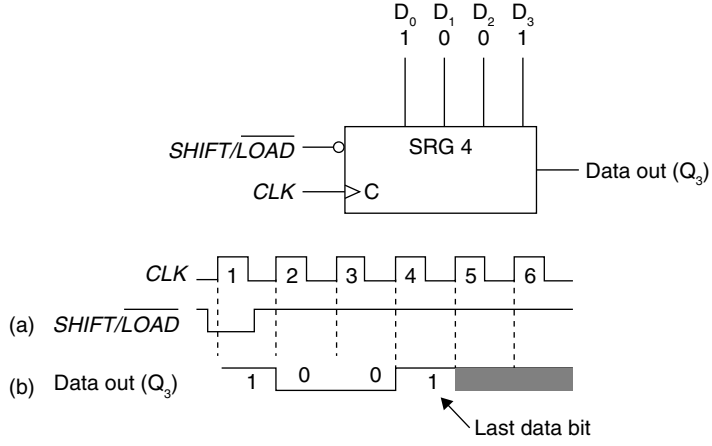


Fig.10.13 4-bit PISO register

74LS165 8-bit Parallel Load Shift Register: The 74LS165 is an example of an IC shift register that has a parallel in/serial out operation. A logic low on the SHIFT/LOAD input (SH/LD) enables all the NAND gates for parallel loading. When an input bit is 1, the flip-flop is asynchronously SET by a LOW out of the upper gate. When the input bit is 0, the flip-flop is asynchronously RESET by a LOW out of the lower gate. It can perform the operation of serial in/serial out register by entering the data serially on the SER input. The clock pulse can be controlled by keeping the logic high at the CLK INH input. The serial data outputs of the register through terminals Q_7 and its complement Q_7. This method is different from the previously discussed patterns thus, demonstrating that the registers can be used in several ways to accomplish the same function. The logic symbol, block diagram and timing waveform is given in Figure 10.14.

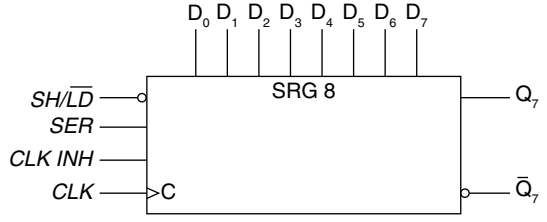


Fig.10.14(a) Logic symbol 8-bit PISO register

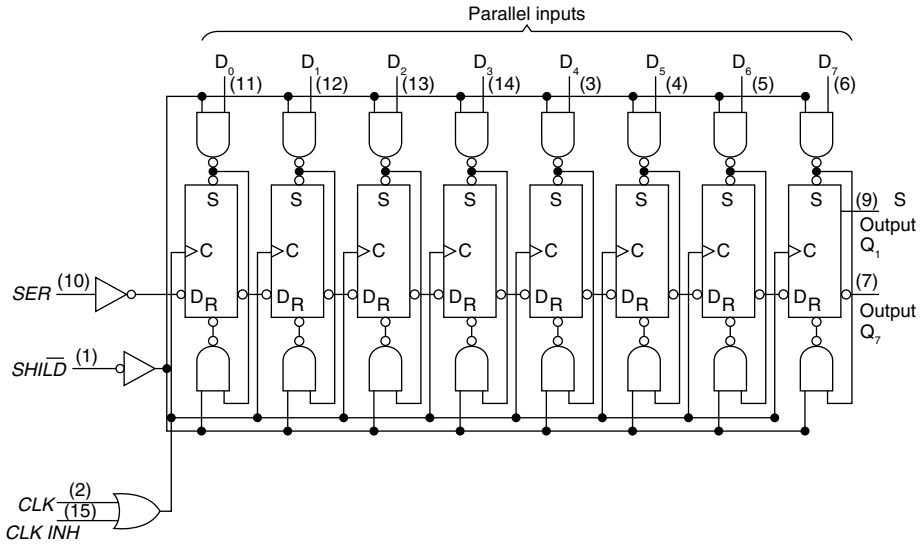


Fig.10.14(b) Logic diagram

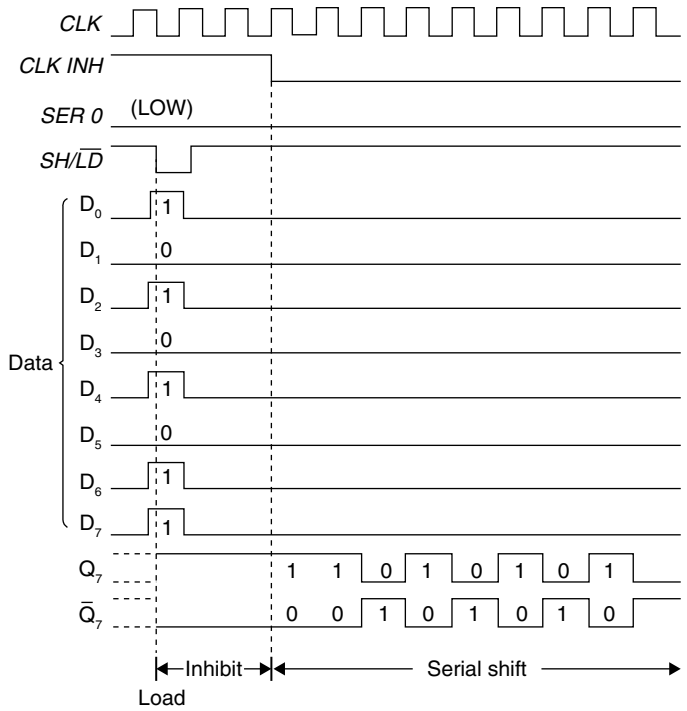


Fig.10.14(c) Timing waveform

Parallel In/Parallel Out Shift Register (PIPO): The parallel in/serial out shift register can also be used as PIPO by taking the output, while the parallel output enable line is HIGH. The data can enter and leave the register at the same time (Figure 10.15).

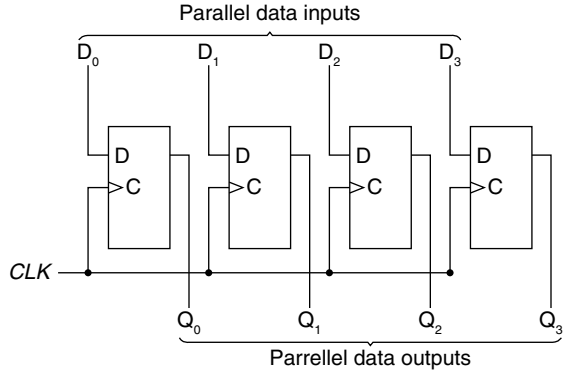


Fig.10.15 PIPO register

The 74LS195A 4-bit Parallel Access Shift Register: The 74LS195A can be used for parallel in/parallel out operation, serial in/serial out and serial in/parallel out operations. The output Q_3 is used for parallel in/serial out operation. The logic block symbol is shown in Figure 10.16.

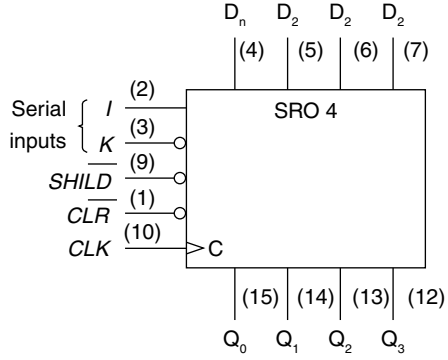


Fig.10.16 Block symbol 4-bit PIPO register

When the clock is high and the SHIFT/LOAD input (SH/LD) is low, the data on the parallel inputs are entered synchronously. When SH/LD is high, stored data shifts from right (Q_0 to Q_3) synchronously with the clock. Inputs J and K are the serial data inputs to the first stage of the register (Q_0); Q_3 can be used for serial output data. The active-low clear input is asynchronous.

The timing diagram as shown in Figure 10.17 illustrates the operation of this register.

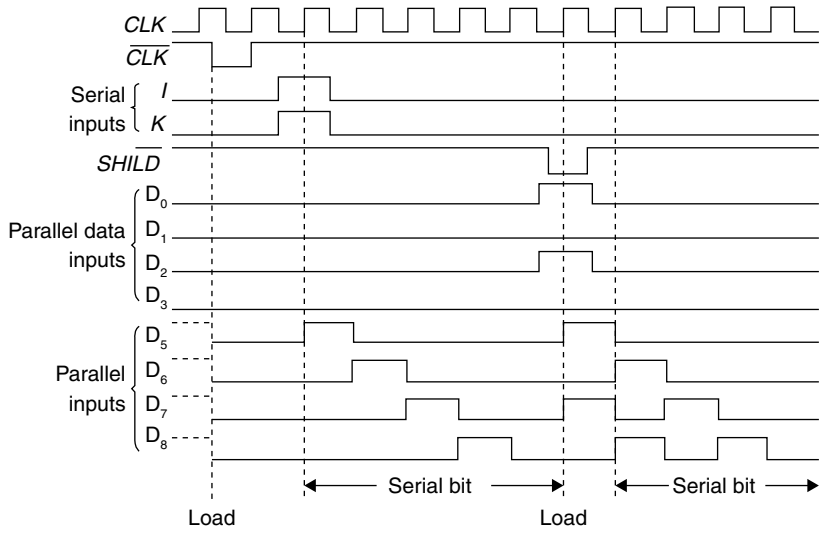


Fig.10.17 Timing diagram of 4-bit PIPO register

BIDIRECTIONAL SHIFT REGISTERS

In a bidirectional shift register, the stored data can be shifted either from left to right or right to left. Two separate input terminals are present which are used for serial input to be shifted to right to left and one for data to be shifted left to right by changing the shift. Inputs between logic levels the direction in which the data is shifted can be determined. The logic level for the data to be shifted to right and data to be shifted left are complementary to each other. When shift right is at logic 1 the top most AND gates are enabled. This is given to the first flip-flop A, the output of flip-flop A is connected to AND gates and whose output is connected to D terminal of flip-flop B and so on. This will shift the serial input data to the right as shown in Figure 10.18.

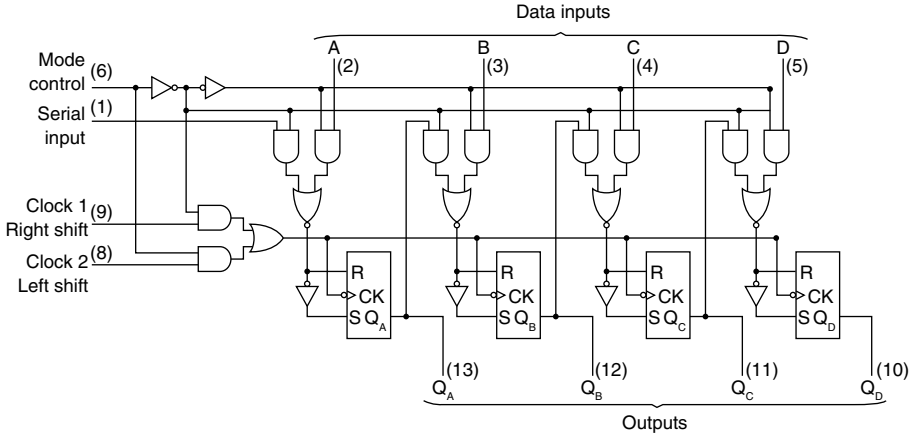


Fig.10.18 Bidirectional shift register

Conversely when shift left is at logic 1, the lower AND gates are enabled and Q_D is connected to the D terminal of flip-flop C, Q_C to ff B, and so on. The circuit now shifts the serial input data from left to right. The 74LS194A is an example of a universal bidirectional shift register.

Mode Control High

All AND gates 2 are enabled while AND gates 1 are disabled. The data is loaded into the *parallel input.*

Mode Control Low

The parallel loaded data is shifted to the right for every clock pulse as the AND gates 1 are enabled and gates 2 are disabled. A serial data entered at AND gate is loaded and shifted. For shifting the data right to left, the serial input is connected to D input of AND gate D_2 and Q_D to D_C, $Q_{C\,to}$ D_B, and Q_B to D_A. When the mode control is high, the data is shifted right to left. Two clock inputs are provided, one for shifting data left to right and other for right to left.

A bidirectional shift register has both serial and parallel input and output capability. The parallel input is synchronous with a positive transition of the clock. The data is loaded to the parallel inputs with a HIGH to the S_0 and S_1 inputs. The shifting of data to the right is accomplished synchronously with the positive edge of the clock when S_0 is HIGH and S_1 is LOW. The serial data is entered at the terminal SR SER. When S_0 is LOW and S_1 is HIGH, data bits shift left synchronously with the clock, and new data are entered at the shift-left serial input SL SER. Input SR SER goes into the Q_0 stage, and SL SER goes to Q_3.The sample timing diagram is shown in Figure 10.19.

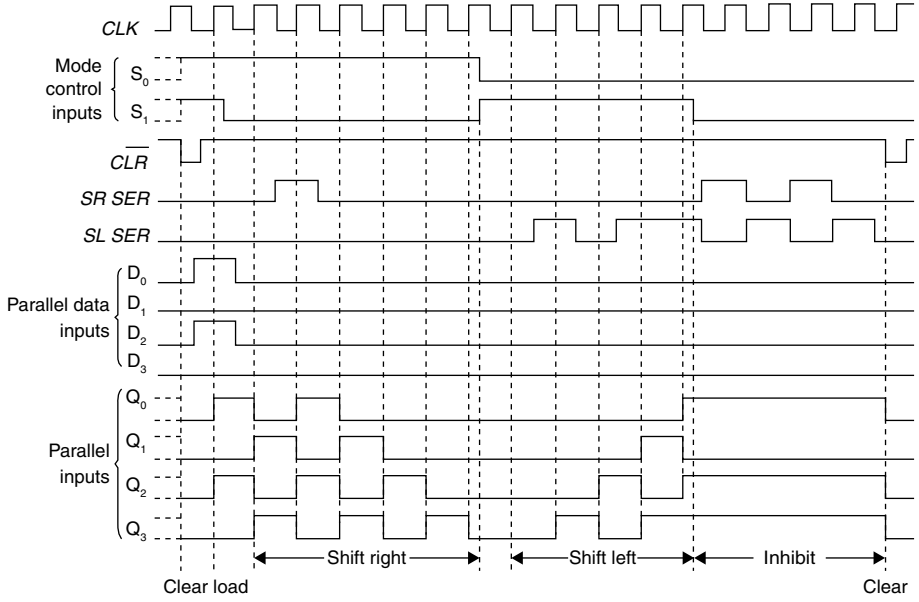


Fig.10.19 Timing diagram bidirectional shift register

UNIVERSAL SHIFT REGISTER

A register in which entering and removing data by all four methods (SISO, SIPO, PIPO, PISO) besides shifting the data from left to right or right to left is called as a universal shift register. IC 7495 is a 4-bit universal shift register. The logic symbol is shown in Figure 10.20.

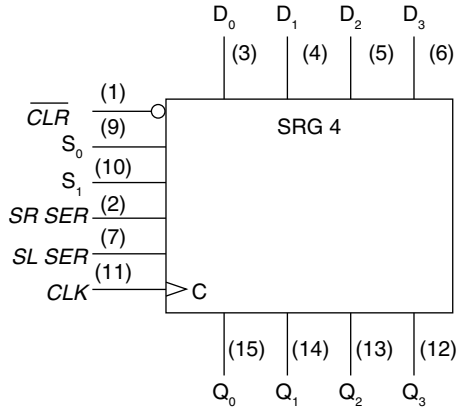


Fig.10.20 Logic symbol of 4-bit universal shift register

It has 4 input and the corresponding outputs Q_A, Q_B, Q_C, and Q_D are taken to be low when rest is 0 irrespective of the other inputs. When clock = 0, the outputs are in the previous state S_1 and S_0 however determine the output. $S_0 = S_1 = 1$, the register works in PIPO mode. The data at the serial input is avoided and only the data at parallel input will be considered. This data will appear at the corresponding parallel outputs for positive edge trailing of the clock pulse. $S_0 = 1$, $S_1 = 0$, the shift serial right input is activated for every leading clock pulse the existing data is shifted to the successive stage at the right and the data at Q_D will be lost. This mode has no effect on the parallel input data. $S_0 = 0$, $S_1 = 1$ the shift serial left input is activated; the existing data shifts one stage to the left. The data at the output Q_A of the flip-flop moves out of the register. Parallel inputs have no effect. $S_0 = S_1 = 0$ when both are low then the outputs remain in their existing states and has no change. This is similar to the hold state.

SHIFT REGISTERS APPLICATIONS

Ring Counters

The basic principle of using shift registers as counters is its use as a feedback loop. The normal output from the previous register is fed to the input of the successive register. Hence it works as SISO register.

Consider N stages at the rate of 1-bit per stage. The feedback is given from the Nth stage to the first stage. When a data "1" is loaded to the left-most register with all other

registers at 0, then the loaded data "1" gets shifted right serially for each clock pulse. To enable decimal counting 10 shift register is required each indicating the decimal value corresponding to bit 1 on it. In this way, a decoder circuit can be avoided. This circuit proves inefficient as it requires 10 flip-flops, but a decoder circuit with a ripple counter is much complex.

Since the counter goes through N states before repeating the sequence and is also called as MOD-N or count by N counter. When the data is shifted from left to right it is also called as % by 2^N counter and while shifting right to left it is a scalar by N (multiplied by 2^N).

Consider a 3-stage ring counter. The left most significant bit state has 1 with 0 in the other two states (100). When the first clock pulse is applied, the shift register reads 010 or a decimal equivalent of 2. Hence the count is divided by 2 for each clock pulse.

The clock frequency is divided by N for a 1 at the right most last flip-flop. The above shift register is not self-starting therefore a clock pulse is required to clear the states and a clock pulse to preset inputs. Since a ring counter keeps repeating a sequence in a ring form it is also known as the circulating shift register. The transition diagram for 3-state ring counter is shown in Figure 10.21.

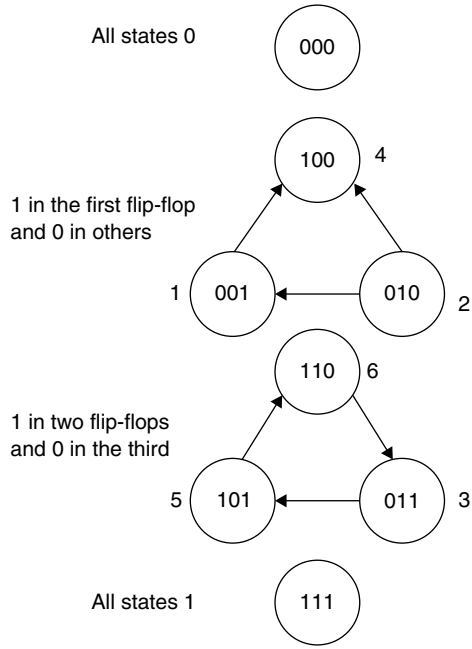
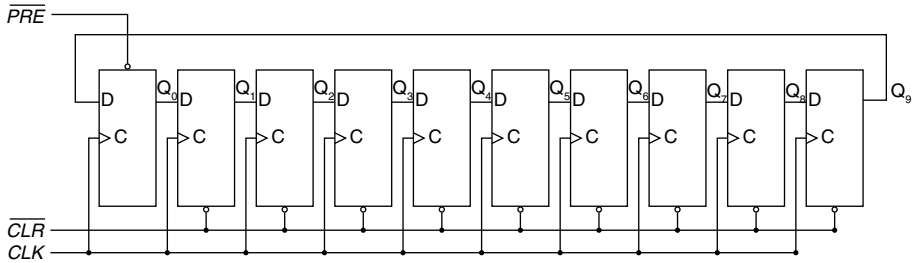


Fig.10.21 Transition diagram of 3-bit ring counter

A logic diagram and sequence for a 10-bit ring counter is shown in Figure 10.22. Initially, a 1 is preset into the first flip-flop, and the rest of the flip-flops are cleared. Here the output Q instead of its complement is applied as feedback to input D of the first flip-flop.



Clock pulse	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

Fig.10.22 Logic diagram and sequence of 10-bit ring counter

Johnson Counter

The Johnson ring counter is similar to the ring counter and is also known as the "Twisted Ring Counters" or "Switch Tail Ring Counters". The complementary output of the last stage is connected to the J input of the first stage or flip-flop and the output Q is connected to the K input of the of the first JK flip-flop. A Johnson counter can also be constructed using a D flip-flop. The complemented output of the last stage is connected to the D input of the first JK flip-flop. Since the connections are done from the complementary terminals to the inputs and also the output is measured in the last or tail stage, it is called switch tail ring counters. In some cases, the outputs can be measured from intermediate stages.

5-bit Johnson Counter: The difference between a ring counter and Johnson counter is the feedback connection from the output to the input. This type of counter reduces the number of flip-flops used in the circuit by half when compared to the standard counters. A counter with N states will generate a sequence of 2N different states and hence saving on 100% in the number of flip-flops. Thus, the modulus of a Johnson counter is given as 2N. The circuit of a 5-bit Johnson counter is shown in Figure 10.23.

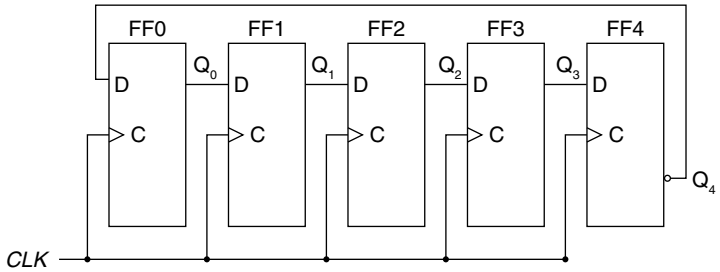


Fig.10.23 5-bit Johnson counter

Consider a decade Johnson counter that has five states. The state table is given in Table 10.1. The 5-stage Johnson counter has $2 \times 5 = 10$ states. The inverted output is fed to the input which causes it to count different pattern. The counter instead of following the usual pattern of counting for a 5-bit counter, the Johnson counter counts up and then down as the initial logic "1" passes through it to the right replacing the preceding logic "0". A 5-bit Johnson ring counter passes five logic "0" and then five logic "1" thereby producing a 10-bit pattern.

Table 10.1 State table of decade Johnson counter

Counter state after clock pulse	Q_A	Q_B	Q_C	Q_D	Q_E
0	0	0	0	0	0
1	1	0	0	0	0
2	1	1	0	0	0
3	1	1	1	0	0
4	1	1	1	1	0
5	1	1	1	1	1
6	0	1	1	1	1
7	0	0	1	1	1
8	0	0	0	1	1
9	0	0	0	0	1
10	0	0	0	0	0

Thus, the counter fills with logic 1's in the first half operation of the shift register and 0's in the second half of the register. As the inverted output Q that is connected to the input D, it provides a feedback mechanism. This 10-bit pattern continually repeats. The timing diagram for a 5-bit Johnson counter is shown in Figure 10.24.

It is evident from the waveform that the output of each state is a square wave whose time period is 10 times the clock pulse, whereas the waveform of a ring counter is only 5 times the clock pulse and hence not a square wave. The output of each flip-flop is shifted by a single clock pulse. The truth table of the Johnson counter gives

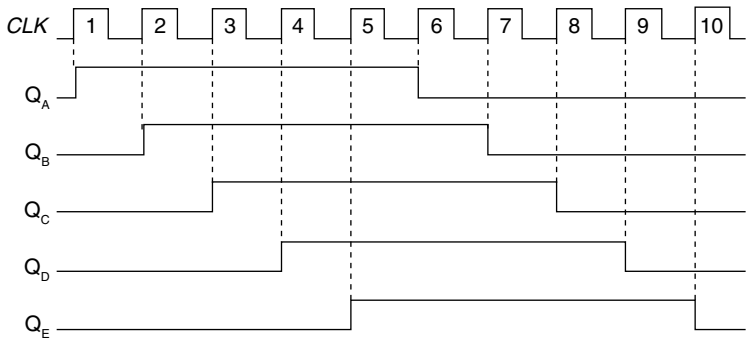


Fig.10.24 Timing diagram of 5-bit Johnson counter

two inferences. Firstly from state 0 to state 9 the decimal equivalents of the states are given as 0, 1, 3, 7, 15, 31, 30, 28, 24, 16, 0. Second, the decimal equivalents are not in any particular order and hence decoding is done on assumption. Except for the first state S_0 no other state has both A and E as 0.

$$A'E' = 0$$

Also only in state 1 a is 0 and b is 1.

AB' = 1 similarly, BC' = 2, CD' = 3, DE' = 4, AE = 5, A'B = 6, B'C = 7, C'D = 8, D'E = 9.

Thus, a 2-input AND gate can decode the counter. There exists certain unused state s in Johnson counter that are given by $2^N - 2N$. Thus, the unused state of a decade counter is $2^5 - (2 \times 5) = 22$. From the above state table, the decimal counts 2, 6, 10, and 14 are unwanted states. Hence, the function given as F = A'D'E is used to avoid the undesired states and the counter enters the sequence in a duration of 10 clock pulses.

Johnson counter can be enabled to count in odd modulus ($2N - 1$). For odd counting, the feedback connection is from the output of the last but one stage to the input of the first stage. The diagram is shown in Figure 10.25.

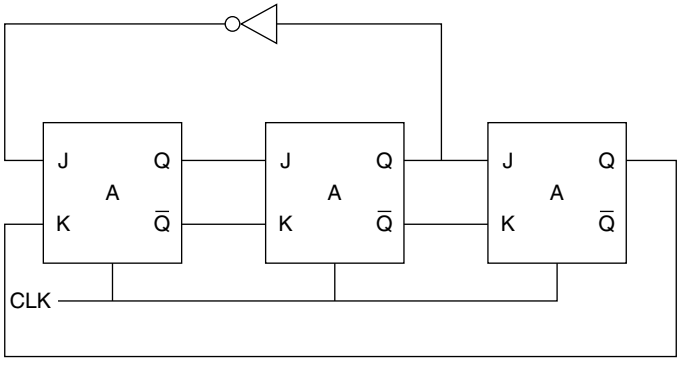


Fig.10.25 Johnson counter counting odd modulus

Odd counting in the above circuit can also be achieved by connecting the complemented output of the last stage Q_C to J_A and Q_B to K_A. This enables the counter to self-start mode. The standard 5-stage Johnson counter such as the commonly available CD4017 is generally used as a synchronous decade counter/divider circuit. Johnson ring counters are available in standard TTL or CMOS IC form, such as the CD4017 5-Stage, decade Johnson ring counter with 10 active HIGH decoded outputs or the CD4022 4-stage, divide-by-8 Johnson counter with 8 active HIGH decoded outputs.

Sequence Generator

Shift registers can work as sequence generator when they are provided with feedback. They find their applications as counter, frequency divider in timing circuits, code generators, etc. Every sequence has a length and this is defined as the number of bits before the sequence repeats.

For example, a sequence 1 – 0 – 1 – 1 – 0 – 1′– 1 – 0 – 1 – 1 – 0 – 1 has a length of 4-bit (101). Based on the length of the sequence, the number of flip-flops required is decided. The number of states or flip-flops and length l of the bits are related by the expression:

$$\text{Length (l)} \leq 2^N$$

Consider an 8-bit sequence 100100100 and the flip-flops required will be 3 since $2^3=8$. The maximum length sequence generator is given as $l \leq 2^N - 1$. A 4-bit shift register has 15 states and the 16th state being 0000 will block the shift register and will not allow it to come out of this state. The number of flip-flops required is 4 as $2^4 = 16 - 1 = 15$.

The sequence table for the sequence 100110101111000 is given in Table 10.2.

Table 10.2 Sequence table

State/Clock pulse	Q_a	Q_b	Q_c	Q_d	$F(Q_a, Q_b, Q_c, Q_d)$
S_1	1	0	0	0	0
S_2	0	1	0	0	0
S_3	0	0	1	0	1
S_4	1	0	0	1	1
S_5	1	1	0	0	0
S_6	0	1	1	0	1
S_7	1	0	1	1	0
S_8	0	1	0	1	1
S_9	1	0	1	0	1
S_{10}	1	1	0	1	1
S_{11}	1	1	1	0	1
S_{12}	1	1	1	1	0
S_{13}	0	1	1	1	0
S_{14}	0	0	1	1	0
S_{15}	0	0	0	1	1

The sequence can be written after each clock pulse. The outputs Q_b, Q_c, and Q_d are started after a delay by one clock pulse. F is found from the next state of Q_a. The state of the shift register depends on all the flip-flops hence repetition in the states are avoided. The Karnaugh map for the above sequence is given below.

$$C'D + CD' = C \oplus D$$

AB \ CD	00	01	11	10	
00					
01	1	1	1	1	
11					
10	1	1	1	1	

Hence the sequence generator is constructed using 4 flip-flops and a XOR gate as shown in Figure 10.26.

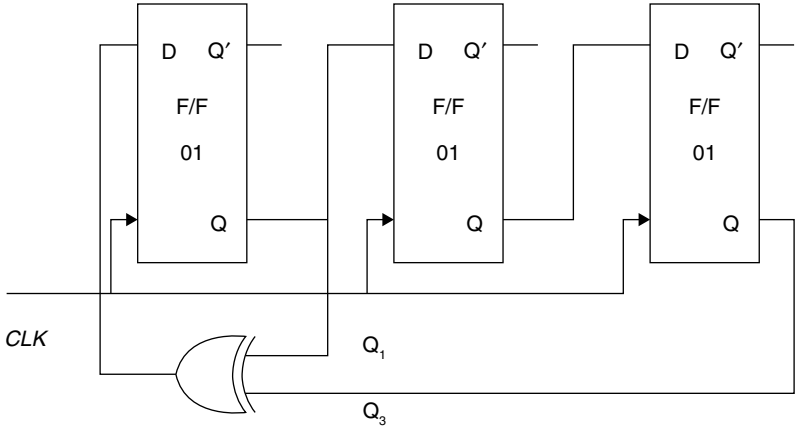


Fig.10.26 Sequence generator circuit

To avoid the state 0000, the circuit can be modified by taking the complement of the state. This is achieved by having a network of AND gates. The sequence generator for a non-maximal length is similar to the maximal length. In this case, repetition of states is unavoidable and leads to uncertainty of the generator. This can be overcome by giving additional clock pulses. A feedback logic is determined in this case. Another type of sequence generator is the random sequence generator which starts the sequence from the output of any one flip-flop. The cycle repeats for every $2^N - 1$ clock pulses. The randomness increases with the increase in the stages in the register.

If $N = 5$, then the number of clock pulses is given as $2^N - 1 = 2^5 - 1 = 32 - 1 = 31$. For a clock frequency of 10MHz, the sequence repeats itself for a time period given as $31/(10 \times 10^6) = 0.3$ μsec.

Time Delay

The shift registers introduce a natural one clock delay between successive registers. Therefore, if there are N state shift registers then the total time delay is given as NT_C, where T_C is the time period of the clock pulse. Thus, the amount of delay can be controlled by the number of stages in the register and the clock frequency. A clock frequency of 1 MHz has a period of 1 μs. The serial in-serial out shift register can be used as a time delay device. The block symbol and timing diagram is shown in Figure 10.27(a) and (b).

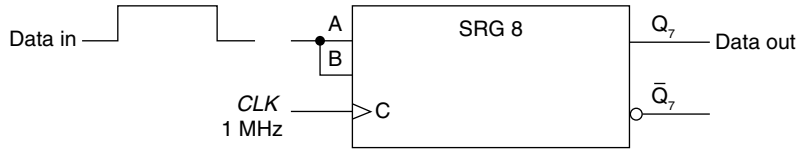


Fig.10.27(a) Time delay circuit using SISO register

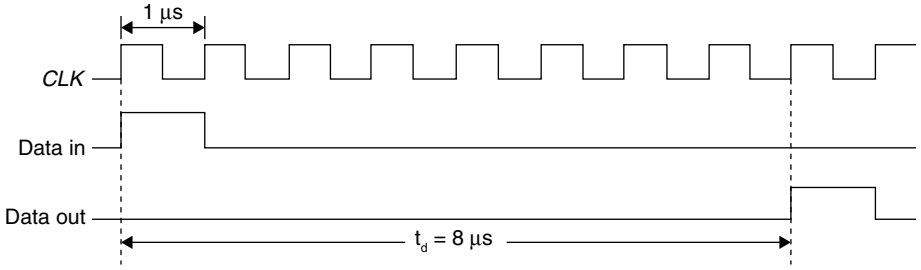


Fig.10.27(b) Timing diagram

Serial to Parallel Conversion or Vice Versa

Most of the computer or microprocessor-based system requires that their input data is in parallel format but these devices get their data from external devices that send or receive serial data. Therefore, the data is to be converted into the parallel form before it is fed to the computer systems. Thus, a serial-to-parallel conversion is required. The converse of this is also possible and true. The circuit for serial to parallel conversion is shown in Figure 10.28.

Universal Asynchronous Receiver Transmitter (UART)

The UART is a universal asynchronous receiver/transmitter. UART is a chip that controls computer serial ports, disk drive interrupts, screen refresh cycles, and other devices that require timing. UART converts the bytes into serial bits and transmits the bits through an asynchronous transmission. It removes the start and stop bits for each

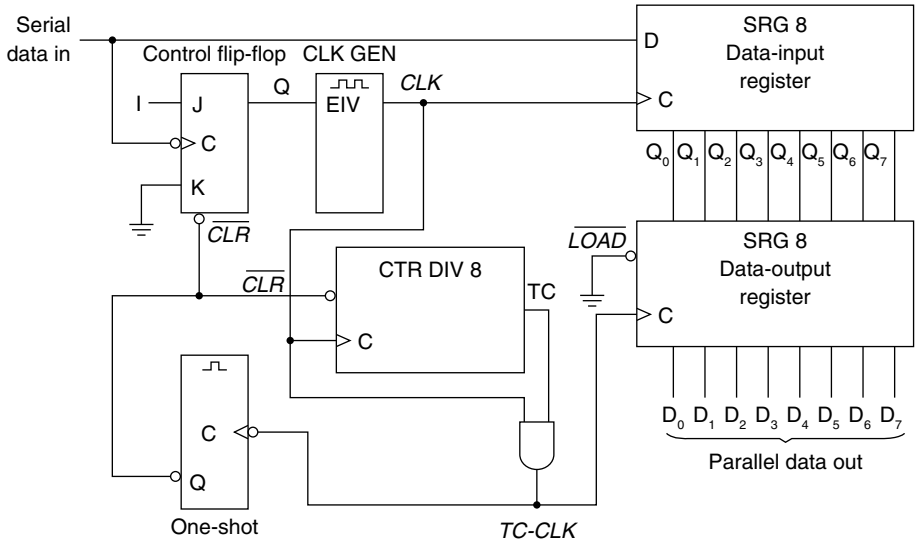


Fig.10.28 Serial to parallel conversion

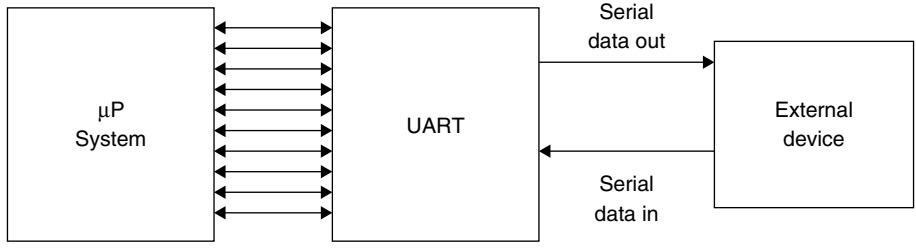


Fig.10.29 UART block diagram

character. Thus, UART converts serial data to parallel data or parallel data to serial data. The block diagram of a UART is shown in Figure 10.29.

The speed at which the data is transmitted on a line is called baud rate. Baud is defined as bits per second. Example if 10 bytes are sent per second on the line, the baud rate is 10×8 bits = 80. If along with the bytes, start, parity, and stop bits are also sent in that one second, then the baud rate increases.

MULTIPLE CHOICE QUESTIONS

1. A register is used to
   (a) Store data
   (b) Count data
   (c) Provide pulse
   (d) None of the above

2. Which of the following register is used to shift one bit of data in and out of the register serially at a time?
   (a) SIPO (b) PISO
   (c) SISO (d) PIPO
3. Which of the following is not a shift register?
   (a) SIPI (b) SIPO
   (c) PISO (d) PIPO
4. When clock pulses are applied, a register
   (a) Counts the data (b) Doubles the data
   (c) Shifts data towards left or right (d) None of the above
5. A universal shift register is one that
   (a) Enters data in parallel and serial direction
   (b) Shifts data either to left or right
   (c) Bidirectional serial load
   (d) All of the above
6. In a shift register when the outputs are connected together then
   (a) Parallel data is taken out (b) Serial data is taken out
   (c) All outputs are 0 (d) All outputs are 1
7. When all the flip-flops of a shift register are preset with data then it is the _____ of shift register.
   (a) Serial load (b) Parallel load
   (c) High load (d) Low load
8. When one flip-flop is preset and all others are cleared then register functions as
   (a) Universal shift register (b) Ring counter
   (c) Johnson counter (d) All of the above
9. A 4-bit Johnson counter has _____ states.
   (a) 2 (b) 4
   (c) 8
10. An 8-bit parallel out shift register is initially cleared and the bit sequence 10011001 is serially entered (right-most bit first). What are the Q outputs after three clock pulses?
    (a) 10011001 (b) 00000000
    (c) 11111111 (d) 11001000
11. A 4-bit serial in/serial out shift register is initially clear and data 0011 is entered serially entered (right-most bit first).The output after two clock pulse is
    (a) 1100 (b) 0011
    (c) 0000 (d) 1111

12. Pulses of equal width are generated using
    (a) Johnson counter (b) Flip-flop
    (c) Ring counter (d) Clock
13. In a 4-bit SIPO shift register is the bit sequence 0001 is entered then the output at the end of three clock pulse is
    (a) 0000 (b) 0010
    (c) 1111 (d) 1000
14. A 4-bit ring counter has an initial state 0010, the output after the third clock pulse is
    (a) 1111 (b) 0000
    (c) 0010 (d) 1000
15. To load data serially into a 3-bit shift register _____ clock pulses are required.
    (a) 2 (b) 3
    (c) 4 (d) 6
16. A parallel in/parallel out shift register has data 1100. The output after two clock pulses is
    (a) 1100 (b) 1100
    (c) 1000 (d) 0000
17. Johnson counters are
    (a) Synchronous counters (b) Ring counters
    (c) Asynchronous counters (d) None of the above
18. The data loading in shift registers are controlled by
    (a) Power supply (b) High outputs
    (c) Strobe signal (d) Clock pulses

Answers

1. (a) 2. (c) 3. (a) 4. (c) 5. (d) 6. (a) 7. (b) 8. (b)
9. (c) 10. (d) 11. (a) 12. (c) 13. (d) 14. (b) 15. (b) 16. (d)
17. (c) 18. (d)

DESCRIPTIVE QUESTIONS

1. Explain the two important operations of registers.
2. Discuss the various applications of registers.
3. Explain the universal shift register with the necessary circuit diagram.
4. Draw and explain the working of a PISO shift register using D flip-flop and hence discuss how data 1001 is shifted after two clock pulses.
5. Draw and explain a 3-bit Johnson counter.

6. Draw a 4-bit serial in parallel out register. Also explain how the data 1100 is shifted into the register and the output state at the end of three clock pulses. Draw necessary waveforms.
7. Explain the working of ring shift that has a data 1010 stored in it. What is the output of the ring counter at the end of 3 clock pulses?
8. Write short notes on
   (a) Sequence generator
   (b) Time delay circuit using registers
   (c) Serial to parallel data conversion using registers
   (d) Universal asynchronous receiver transmitter (UART)

COUNTERS

Counters form an indispensable component of digital equipments. Counters quantify the frequency, and the inverse time period. They are used as frequency dividers where the frequency of a given pulse waveform is divided. Counters are sequential circuits that count the number of pulses given to the input terminal. Their clock pulses can be either in binary code or BCD form. The main properties of a counter are timing, sequencing, and counting.

Counters work in two modes.

1. UP counter = EE reads in ascending order from 0 to 1
2. DOWN counter = EE reads in descending order from 1 to 0

Counters can also be classified as ripple and synchronous counters. MOD-16 is a binary counter, while a MOD-10 is not. Both can be operated either by synchronous or asynchronous mode.

A simple clocked JK flip-flop performs as a counter. Counters are generally obtained by number of JK or D flip-flops since both these flip-flops can be operated synchronously or asynchronously. In case of a synchronous counter, all flip-flops are locked simultaneously, and in asynchronous counters each flip-flop functions in turns at specific pulse. Owing to this reason, synchronous counter operate at much faster speed compared to asynchronous counter.

A T flip-flop can be obtained by connecting both the J and K terminals together and given high 1 for negative edge triggered flip-flop together to the opposite state $Q_{n+1} = \overline{Q}_n$.

JK flip-flop functions as a 2-bit counter corresponding to status 0 and 1. A 2-bit counters constructed using JK flip flop and its timing diagram is shown in Figure 11.1.

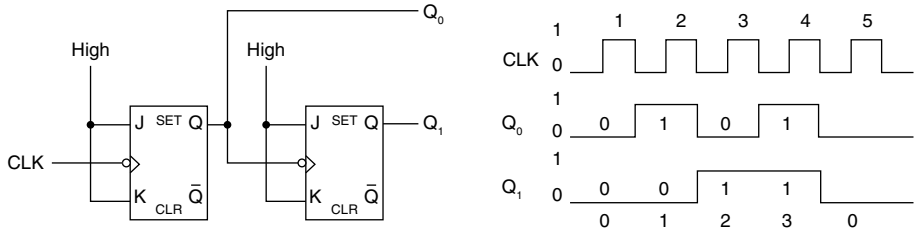


Fig.11.1 2-bit counter and its timing diagram

The counter counts only during the negative edge triggering (NET). For 2-clock pulses we obtain one full pulse cycle and hence called divide-by-2 counter. The counting of a 2-bit UP and 2-bit DOWN counters is shown in Table 11.1.

Table 11.1 Truth table for 2-bit UP and 2-bit DOWN counters

2-bit UP counter		
A	B	Counts
0	0	0
0	1	1
1	0	2
1	1	3

2-bit DOWN counter		
A	B	Counts
0	0	3
0	1	2
1	0	1
1	1	0

If there are N flip-flop then it will be divided by 2^N, Where N is the number of flip-flops. If there are 4 flip-flops the modulus N will be, $2^4 = 16$ or MOD-16 counter. This counter can read values from 0000 to a maximum of 1111 (decimal equivalent =15). The maximum number a counter reads is $2^N - 1$. For a MOD-8 counter the maximum number that the counter can read is decimal 7(111) and resets to 000 at the 9th state. Modulus can also be defined as the number of states the counter changes before resetting to 0 or original state.

When the flip-flops are serially connected, the output of the first flip-flop gives the clock pulse to the next flip-flop then the counter works as asynchronous counter where 1 flip-flop change state one after the other. They are also called ripple counter as the clock pulses pass from 1 flip-flop to the other in forms of ripples.

MOD-3 COUNTER

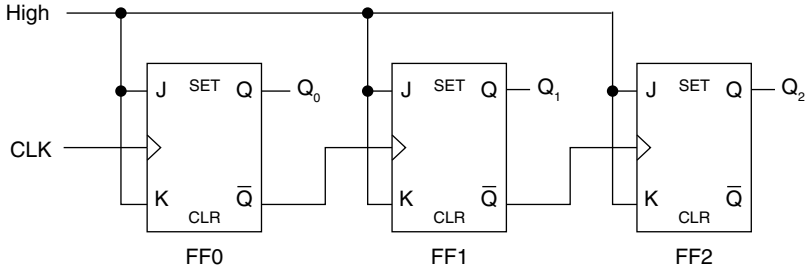


Fig.11.2 MOD-3 counter

A MOD-3 counter using JK flip-flop is depicted in Figure 11.2. The output of flip-flop 0 becomes the clock pulse for flip-flop 1 and output of flip-flop 1 becomes the clock pulse for flip-flop 2. Initially all flip-flops are reset to 0 by feeding high input, i.e $Q_1 = Q_2 = Q_3 = 0$. All flip-flops are negative edge triggered on the trailing edge of clock pulse.

The output of flip-flop 1 toggles to 1 on the negative edge trigger of first pulse. This is sent as the clock pulse input to flip-flop 2 but does not cause any toggle, therefore, Q_2 and $Q_3 = 0$. The count is now 001 or decimal 1.

At the negative edge terminal of the 2nd clock pulse, flip-flop 1 toggles again and Q_1 changes from 1 to 0. This output toggles to flip-flop 2 now $Q_2 = 1$ and $Q_3 = 0$ now the count is 010 or decimal 2. The trailing edge of 3rd pulse toggles flip-flop 1 to 0 to 1 but this transition does not affect flip-flop 2 and Q_2 remains at 1 to give $Q_1 = Q_2 = 1$ and $Q_3 = 0$ the count is 011 or decimal 3. At the end of the 4th pulse, Q_1 toggles to 0 and in turn toggles to $Q_2 = 0$. This change toggles Q_C to 1. Now the count is 100 whose decimal equivalent is 4.

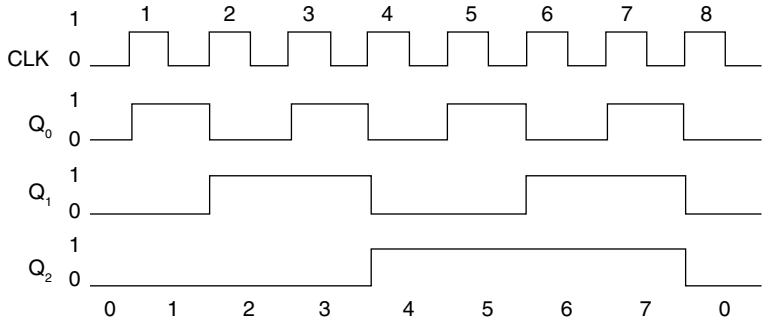


Fig.11.3 MOD-3 counter timing diagram

The process keeps continuing for every clock pulse till the counter counts 101, 110, and 111.

When the counter reaches the count 111 or 7 which being the maximum number, the counter can read all 3 flip-flops are set NET at 2_1 toggles flip-flop 2 so Q_2 changes from 0 and finally this toggles flip-flop 3 thus, making $Q_3 = 0$.

Now the counter is reset to 000 and can begin new counter sequence. The timing diagram of a MOD-3 counter is shown in Figure 11.3.

MOD-16 OR MOD-4 BIT COUNTER

The MOD-4 counter consists of 4 JK flip-flops as shown in Figure 11.4.

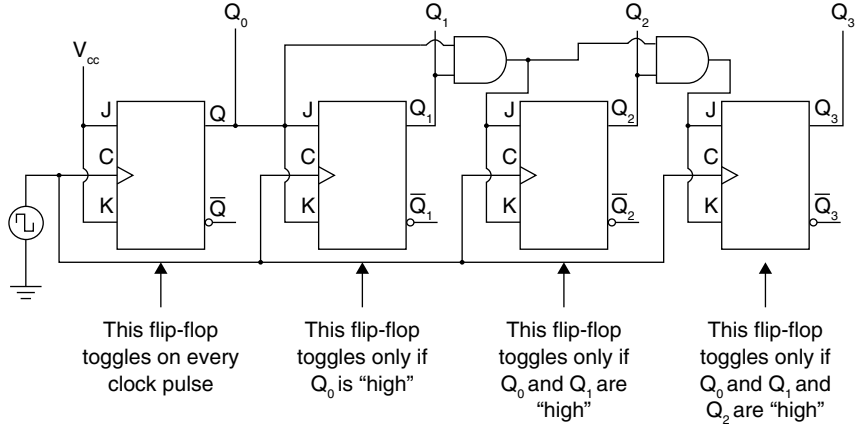


Fig.11.4 4-bit synchronous UP counter

It is a UP counter as the counting is in the ascending order 0 to 15. Initially all flip-flops are reset to 0 by giving high input to R(1) and R(2) of NAND gate. Thus, making $Q_1 = Q_2 = 0$. When $Q_2 = 0$, Q_3 becomes 1 and remains till 8th pulse. Before this Q_1 undergoes two cycles and Q_2 goes through one cycle. At the end of the 3rd pulse, flip-flop A goes to 0 and at NET of the 4th pulse both $Q_1 = Q_2 = 0$ when $Q_2 = 0$ Q_3 becomes 1 remains till 8th pulse. Before this, Q_1 undergoes 2 cycles and Q_2 goes through 1 cycle. At the NET of the 8th pulse Q_1, Q_2, Q_3 move from 1 to 0 and NET of Q_3 toggles flip-flop 4 to 1 and this remains in 1 until the trailing edge or NET of 10th pulse where all outputs Q_1, Q_2, Q_3 and Q_4 reset to 0.

The waveform of MOD-16 counter is depicted in Figure 11.5

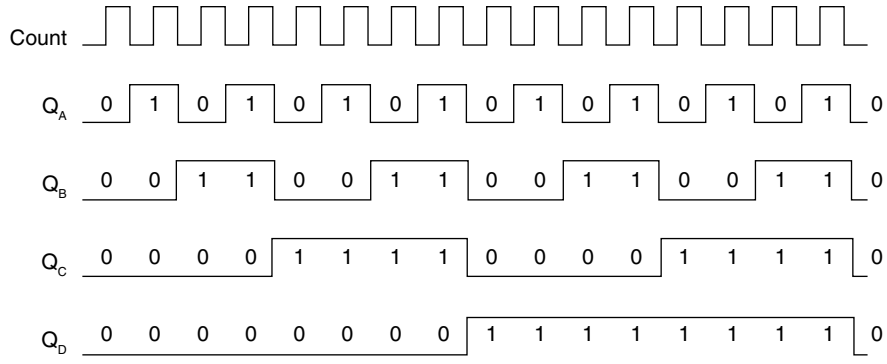


Fig.11.5 Waveform of MOD-16 counter

Thus, to summarize the working of MOD-10 counter, at the start of the trailing edge of 1st clock pulse, flip-flop 1 changes state and all other flip-flops change state in consecutive pulses and move from HIGH TO LOW.

The state diagram of asynchronous counters depicted in Figure 11.6.

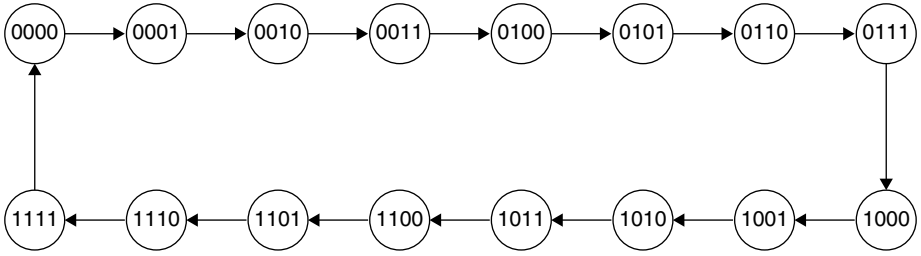


Fig.11.6 State diagram of asynchronous counters

If each flip-flop is 50ns then total propagation delay will be 200ns as there are 4 flip-flops. However, if the clock period is less than 200ns, then flip-flop 1 will set to 1 and can never reset to 0. Therefore, the clock pulse period should be > than $N\ t_{pd}$. The total clock period T is given as

$$T \geq N\ t_{pd} + T_s, \text{ Where } T_s \text{ is the strobing interval.}$$

DOWN COUNTERS

A down counter counts in the descending order, from high to low 1111 to 0000 reducing by 1.

3-bit or MOD-8 Down Counter

MOD-8 down counter works from the right to left, that is, flip-flop 3 first toggles from 0 to 1 followed by flip-flop 2 and flip-flop 1 for consecutive clock pulses. The circuit wave forms and state diagram is shown in Figure 11.7.

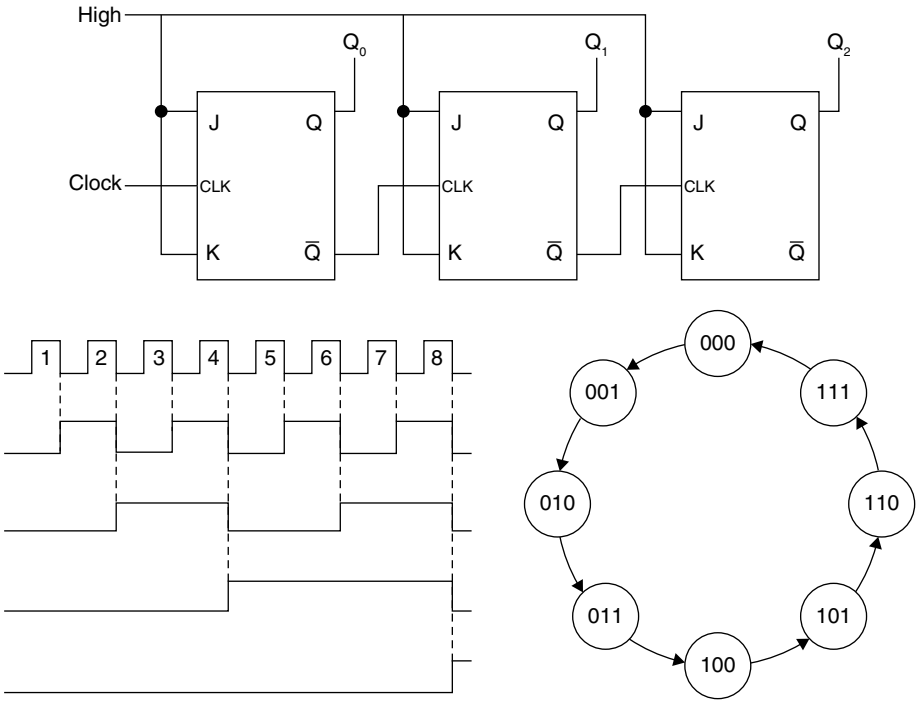


Fig.11.7 3-bit/MOD-8 down counter

DESIGNING OF RIPPLE COUNTER DIVIDED BY *N*

Set of rules to be considered for designing a ripple counter divided by *N*.

- The number of flip-flop from the MOD number is obtained from the equation Modulus $n = 2^N$ where n is the number of flip-flops
- The desired flip-flops are connected serial pattern to f_n as a ripple counter
- The binary number $n - 1$ is formed
- The output flip-flops that are 1 at $n - 1$ are connected to a NAND gate clock pulses are fed to NAND gate

- The output of the NAND gate are connected to preset input of all flip-flops for which Q = 0 at count $n - 1$

For a pure binary, counter slips from 3–5 can be skipped as they are required only for non-binary counters.

DECADE COUNTER OR MOD–10 RIPPLE COUNTER

Now to design a MOD-10 counter or a decade counter as it is commonly known, the following steps are to be performed.

- $2^{N-1} < 10$, where $N = 4$ since the higher power is to be taken. Therefore, 4 flip-flops are required.
- Modulus n of the counter is given as =10 or 1010 and 0101 from left to right.
- The maximum decimal the counter can count is n – 1 = 9 or 1001 binary. Therefore, one count should be skipped.

The output of flip-flop is connected to NAND gate along with the clock. The output from the NAND gate goes to preset terminal of Q_2 and Q_3. In a decade counter, once maximum decimal 9 is reached, the counter does not recycle to 0000 immediately because of the problem delay caused by the NAND gate. But after the delay it toggles from 1001 (decimal 9) to 1010 (decimal 10) thus, making Q_2 to produce glitch or spike. A glitch is an unwanted wave in the waveform.

BCD COUNTER

Binary to decimal counter is weighted in binary whose summing gives the decimal equivalent. The BCD counter counts any decimal value. This counter can be constructed by calculating the number of flip-flops using the relation $2^{N-1} < n < 2^N$ where N = number of flip-flops and n is modulus. Then the flip-flops are connected to form a ripple counter and previous steps are performed. The desired binary count can be counted. Higher counter may be obtained by cascading two or more counters. The new MOD number will be a product of MOD of the individual counters.

E.g. MOD-10 * MOD-20 = 200

For MOD-10, four flip-flops are required and the output of 4th flip-flop is connected to the clock input of MOD-20 counter and the final output of 5th flip-flop will be equivalent to MOD-20.

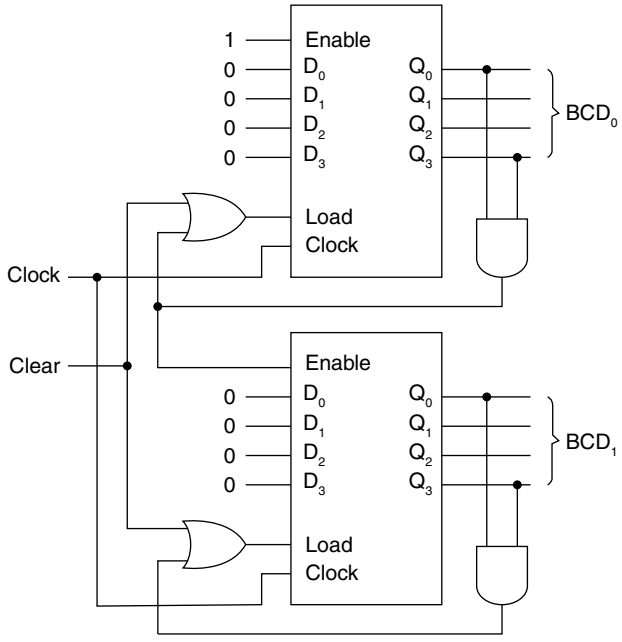


Fig.11.8 Two-digit BCD counter

Consider a two-digit BCD counter as shown in Figure 11.8. It consists of two modulo 10 counters, one for each BCD digit. It is necessary to reset the four flip-flops of the first MOD-10 counter after the count of 9 has been obtained. Thus, the Load input to each stage is equal to 1 when $Q_3 = Q_0 = 1$, which causes 0s to be loaded into the flip-flops at the next positive edge of the clock signal while keeping the enable signal for BCD_1 low at all times except when $BCD_0 = 9$.

The pin configuration in Figure 11.9 shows a BCD counter.

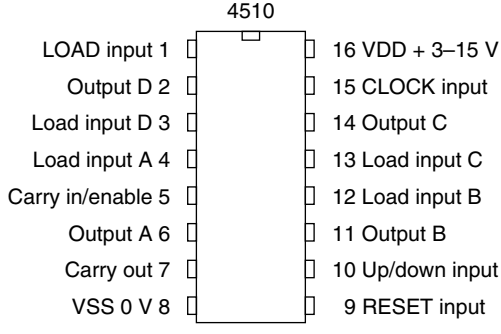


Fig.11.9 Pin out diagram of BCD counter

DECODING A COUNTER

When many flip-flops are cascaded in a counter where, individual outputs can be known by connecting them to a LED. A bright LED is 1 and non-glowing LED is 0. Now each output is connected to LED from the ON or OFF condition of the LED we can predict the decimal and binary code. These LEDs can be controlled by a counter that would commence or stop the operation of the LED.

The individual counter outputs are connected to NAND gates that are further connected to LED or a 7-segment display unit of a BCD counter. The number of NAND gates depend on the modulo of the counter. MOD-4 counter requires 4 NAND gates and MOD-10 counter requires 10 NAND gates. Sometimes all the NAND gate outputs are given to clear terminal to reset but practically all flip-flops do not reset at the same time, one takes longer duration than the other. This can be overcome by connecting a Latch at the output of the NAND gate.

Due to the glitches and spikes caused by propagation delay, the decoder may indicate a false response. To eradicate this problem, all the gates are connected together and kept at LOW (0). This method is known as strobing. Later a HIGH (1) signal is connected to the strobe input and the flip-flops are activated. This method proves to be efficient in controlling the operation of counter but not used in display.

Generally, the last pulse sets the latch to clear the output. Latch remains SET until all flip-flops are cleared. For e.g., MOD-10 counter, the 10th pulse sets the latch to clear the output. The 11th pulse resets the latches such that count is equal to 1 and the trailing edge of the 11th pulse begins a new counting cycle.

UP/DOWN COUNTER

UP/DOWN counter is one which counts backwards, e.g., 543210. There are many applications that require backward counting. A combination of UP and a DOWN counter gives a UP/DOWN counter. It consists of two flip-flops. The first flip-flop works in UP mode and the second flip-flop works in DOWN counting mode. A two-input AND-OR gate sandwiched between the flip-flops behave as a control for the direction of the counter. The circuit of an UP/DOWN counter is shown in Figure 11.10.

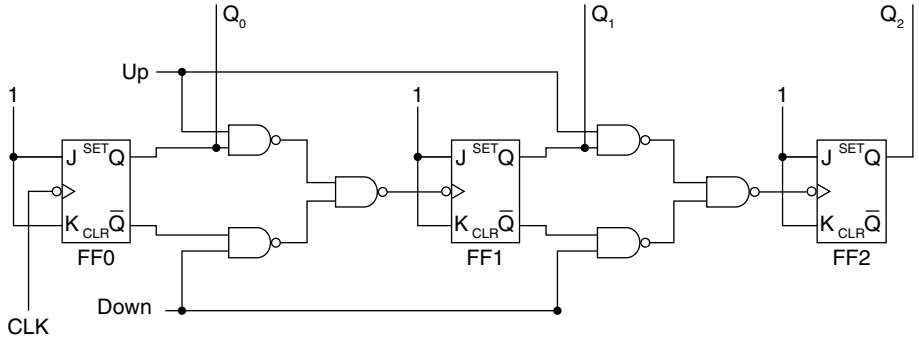


Fig.11.10 UP/DOWN counter

When both flip-flops are at HIGH then the output of the flip-flops are also high. When X = 1 then the output of the first AND gate is HIGH and the second AND gate is 0. Therefore, making the output of OR gate 1. FFA functions as a UP counter having a count 5. During the PET, i.e., low to high Q be 1 X = 0 and X = 1. The output of second AND gate is 1 thereby making flip-flops also 1. The Q outputs of both flip-flops are connected to the pulses. If initially all flip-flops are set then the count is 4. At the negative edge trailing of the first clock pulse first flip-flop resets and Q becomes 1 and the count is 3. The first flip-flop toggles for the next pulse and the negative edge trailing of its Q output resets flip-flop 2 thus, the count is 2. This continues till the two stages are reset. The counter then reads 0 and the next clock pulse counter to begin original counting of 5.

IC 74LS193 and 74HC193 are 4-bit IC UP/DOWN counters and the pin out diagram is shown in Figure 11.11.

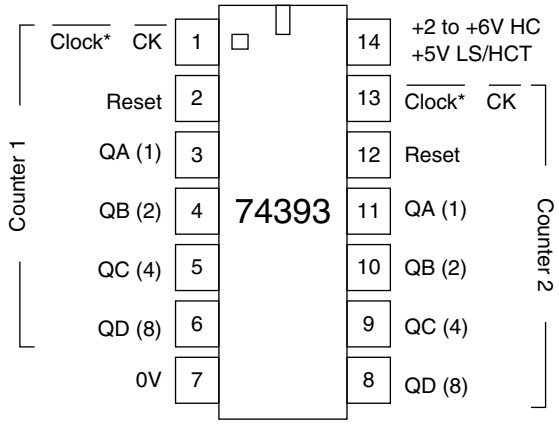


Fig.11.11 4-bit UP/DOWN counter

It consists of two terminals UP for UP counting and DOWN for down counting node. The two clock terminals provided must be maintained at low and active, one for each condition. Clear terminal provided is active high and only for high input it would clear the outputs. By adjusting load pin to LOW and assigning values at the input terminals 1. 10. 15, 19 the counter can start counting from any range. For a UP counting mode, the load input is connected to logic 0 which disables the counter while it is loading information. Then the down input is connected to logic 1. The counter is set to desired range by assigning data word. Load pin is set to logic 1. The input waveform is connected to count UP terminal. If the UP terminal is connected to logic 1 then the counter starts counting in the down mode.

PARALLEL COUNTER (OR) SYNCHRONOUS COUNTER

The counters that were dealt previously were asynchronous (or) serial counters. Parallel counters are those that are connected in the parallel mode or specifically the clock terminals are connected parallel which initiates the flip-flops to work concurrently. Hence they are called parallel or synchronous counters.

The propagation delay caused by the clock frequency which is a major limitation of the asynchronous counters is overcome with these counters. The flip-flop's response to glitches or spikes is also eliminated in these counters.

The values designated to J and K inputs define the state of the flip-flop. When J = K = 0, the output is in Q state and when J = K = 1, the output complements.

To understand the principle of a synchronous counter, let us consider the state table of a 4-bit MOD-16 counter.

Table 11.2 Truth table of MOD-16 counter

State	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0

The truth table of MOD-16 counter is shown in Table 11.2. It is evident from the truth table that the output Q_A of FFA changes its state for every pulse. Therefore, the inputs of this flip-flop must always be 1 and it toggles for each trailing edge of the clock pulse. The output Q_B of the FFB changes when $Q_A = 1$. In other words, it occurs for odd states 1, 3, 5, 7, etc. The output Q_A is connected to J – K inputs of FFB which makes Q_B toggle whenever $Q_A = 1$. Similarly, Q_C changes when both $Q_A = Q_B = 1$ at states 4. The output of FFA and FFB are connected to JK of FFC. When $Q_A = Q_B = Q_C$ then Q_D toggles to reset. Therefore, the output of FFA, FFB, and FFC are connected to the JK inputs of FFD through AND gate. Thus, from the table it is obvious that only flip-flop toggles or FFB, FFC, FFD simultaneously.

The propagation delay is given as

$$\text{Total delay} = \text{FF } t_{pd} + \text{gate } t_{pd}$$

Synchronous counters can be connected as series carry and parallel carry also called as carry look ahead counter.

Maximum Clock Frequency

The maximum clock frequency for a synchronous counter is given as

Synchronous counter with serial/ripple carry = $1/f \geq t_{pd} + (N - 2)\, t_g$

Synchronous counter with parallel/carry look = $1/f \geq t_{pd} + t_g$

where N = number of flip-flops

t_{pd} = time of propagation delay

t_g = time delay of gates

It is to be noted that though AND/NAND or both are used in both ripple and parallel counter, yet the number of inputs to gates in parallel counter keeps increasing with each flip-flop.

DECADE SYNCHRONOUS COUNTER

The decade synchronous counter is depicted in Figure 11.12. In the state table, first 9 states area are required, the rest being do not care states. When JK inputs of the flip-flop A is maintained at high then $J_A = K_A = Q_A = 1$ thus, this condition toggles the FFA. Now FFB changes its state when $Q_A = 1$ and output Q_D of FFD is 0. Then $J_B = K_B = Q_A Q_D$, here $Q_A Q_D$ are ANDed, and this output is connected to JK inputs of FFB. When $Q_A = Q_B = 1$, FFC toggles state thus, making $J_C = K_C = Q_A Q_B$. Hence $Q_A Q_B$ are ANDed and are connected to JK inputs of FFC. Thus, to toggle the FFD, the outputs of Q_A, Q_B and Q_C of FFS, A, B, and C are to be ANDed and outputs $Q_A Q_D$ are ANDed later they are ORed. AND – OR gate combination is placed between FFC and FFD.

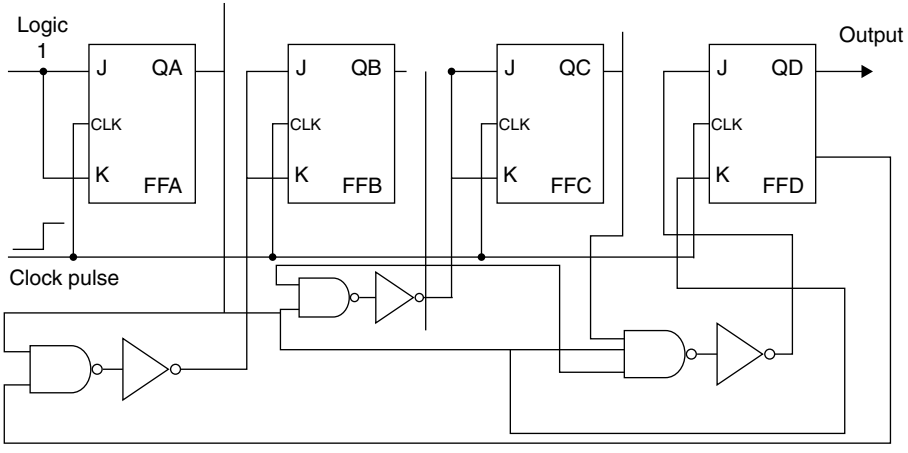


Fig.11.12 Decade synchronous counter

SYNCHRONOUS UP/DOWN COUNTER

A synchronous 3-bit counter is shown in Figure 11.13.

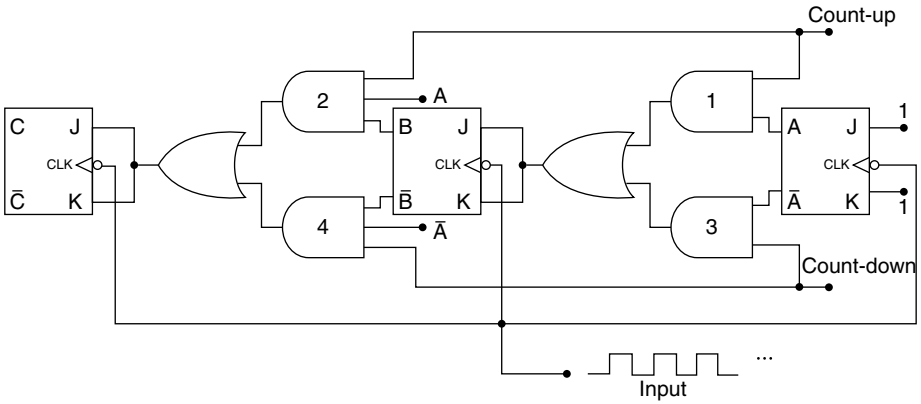


Fig.11.13 3-bit synchronous UP/DN counter

This counter counts in the reverse order, i.e., from 1111 down to 0000 and back to 1111 to begin count. Here the bit in least significant position in the data is complement for the given pulse. The up counter as discussed earlier changes its state when the previous flip-flop at logic 1 and down counter toggles state when the previous state is at logic 0. It is understood from the state table that QA toggles for every pulse between both up and down provided $J_A = K_A = 1$. FFB changes its state when $Q_A = 1$ for UP counting and when $Q_A = 0$ for DOWN counting also

$$J_B = K_B = 1 = Q_A \text{ (UP)} + Q_n \text{ (DOWN)},$$

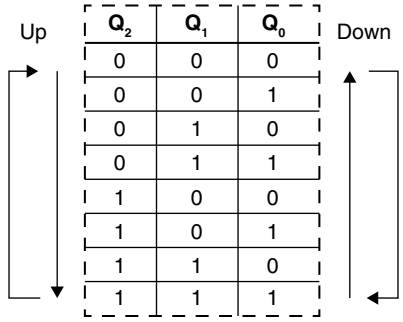
FFC toggles for $$J_C = K_C = 1 = Q_A Q_B \text{ (UP)} + Q_A Q_B \text{ (DOWN)}.$$

This is achieved by placing AND and NOR arrangement in front of each flip-flop. During count up 1 and count down 0 Gate 1 is activated and enables the counter to start up counting. During count up 0 and count down 1, Gates 3 and 4 are activated thus, making counter count down. Thus, by assigning HIGH to count up (or) count down, the counter can start count up in any manner and HIGH can be even during the counting process assigned.

Example 11.1

01234	321	234
UP	DOWN	UP

Solution: But when both up and down inputs of the counter are LOW then NOR gate outputs will be LOW this in turn makes JK inputs LOW thus, disabling the FF from changing states. The state table for an UP/DOWN counter of 3-bit is given below.



Q_2	Q_1	Q_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

IC COUNTERS

Commercially, numerous IC counters are available depending on their applications and mode of functions.

The synchronous counters: 7490, 7492, 7493.

Asynchronous counters: 74190, 74190, 74192, 74193 (UP/DOWN counter) 74161.

A typical counter is shown in Figure 11.14. Cp_1 Cp_0 are the NET click inputs to FF0 and FF1. The first FF0 is not connected and divided by 2. MR_1 and MR_2 are master reset terminals that are internally connected to NAND gate, whose output is connected to clear C_D. All FFS are connected to this clear terminal. In case of non-binary counter, the clear terminal are earthed or connected to corresponding FF outputs.

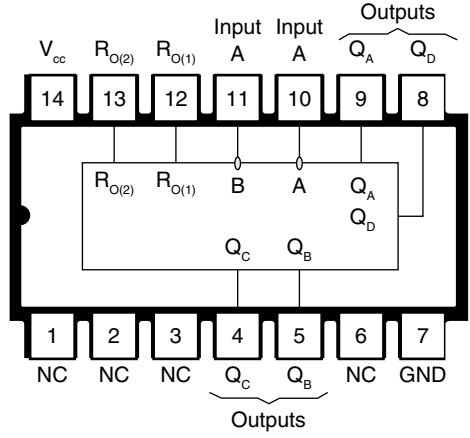


Fig.11.14 Pin out typical counter

Now if the number of flip-flops is to be increased two counters are cascaded and also Cp_1 or Cp_0 are connected to Q_3 depending on the output required.

APPLICATIONS OF COUNTERS

Counters are commonly used in hardware designs as most computations include looping, shift and add multiplication schemes. They are used as clock divider circuits.

ADDITIONAL EXAMPLES

Example 11.2 Construct a MOD-5 ripple counter.

Solution: To construct a MOD-5 counter we need to first calculate the required number of flip-flops.

For MOD-5 $N = 5$

The number of flip-flops $(n) = \log_2 N$

$= \log_2 5$

$= 2.322$

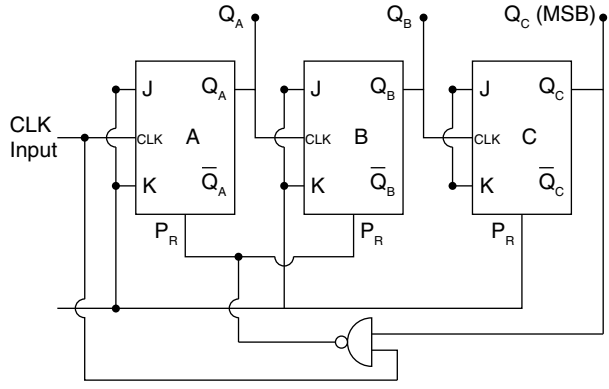
∴ The number of flip-flops is rounded to 3. We require 3 JK flip-flops to construct MOD-5 counter. The JK flip-flops are given negative edge.

The given number $N = 5$

Binary equivalence $= N - 1 = 5 - 1 = 4 = 100.$

where $\underset{1}{Q_C}$ $\underset{0}{Q_B}$ $\underset{0}{Q_A}$. This shows that output Q_C of flip-flop C is maintained at high.

Connections are made as shown in the figure below. The high logic to Q_C is provided. The output of the NAND gate is connected to preset of flip-flop A and B whose outputs are '0'. The JK inputs of the flip-flops A, B and C are given logic high. When the circuit is switched on the counter starts counting states 0, 1, 2, 3, 4. Once it reaches state after 4, it resets itself to 0 and starts counting again from 0. Thus, the counter will reset itself after every 5 clock pulse. This makes the frequency of flip-flop C to one-fifth of the clock



frequency. The MOD-5 counter thus is also known as divide-by-5 counter. The truth table for MOD-5 is shown below.

Time pulse	Count state	Q_C	Q_B	Q_A
t_0	0	0	0	0
t_1	1	0	0	1
t_2	2	0	1	0
t_3	3	0	1	1
t_4	4	1	0	0
t_5	5	0	0	0

Example 11.3 Design a MOD-6 counter by using divide-by-*N* ripple counter.

Solution: To MOD-6 or divide-by-6 ripple counter, the number of flip-flops required is to be measured.

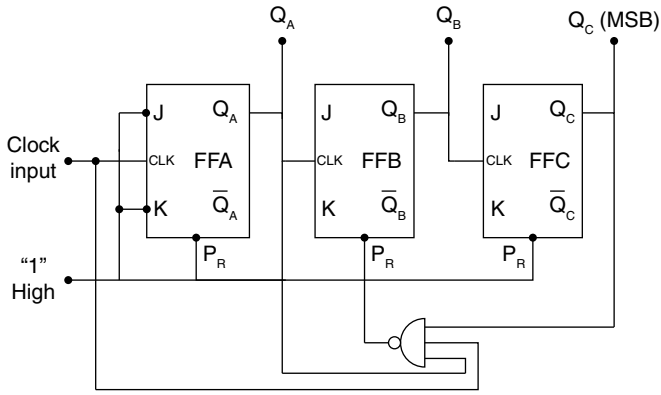
$$N = 6$$
$$\therefore n = \log_2 N$$
$$= \log_2 6$$
$$= 2.585$$

Therefore, the number of flip-flops 2.585 which can be rounded to 3. MOD-6 counter requires 3 JK flip-flops that are negative edge triggered and connected to function as a ripple counter. To determine the number of states, the binary equivalent is to be found.

$$N - 1 = 6 - 1 = 5 = 101$$

Thus, the outputs are $Q_C Q_B Q_A = 101$. It is evident that Q_C and Q_A are given logic high or 1. The output from flip-flop A and flip-flop C are connected as inputs to the NAND gate. The output of NAND gate is connected to preset of FFB. The circuit has six states 0, 1, 2, 3, 4, 5 and it counts from 0 to 5. During the 6th clock pulse, for the positive edge all the flip-flop will preset to 1 and for the negative edge the counter will start counting from 0. This resets the counter. Therefore, 6 clock pulses are required to reset the counter. Therefore, the frequency of output at flip-flop C is 1/6th of the clock frequency.

MOD-6 counter is also known as divide-by-6 ripple counter. The circuit and truth table is shown overleaf.



Time pulse	Count	Q_C	Q_B	Q_A
t_0	0	0	0	0
t_1	1	0	0	1
t_2	2	0	1	0
t_3	3	0	1	1
t_4	4	1	0	0
t_5	5	0	0	0
t_6	6	0	0	0

Example 11.4 Design MOD-12 ripple counter based on the divide-by-*N* ripple counter technique using JK flip-flop. Also draw the truth table.

Solution: The MOD-12 ripple counter is constructed using JK flip-flops based on the divide-by-*N* ripple counter technique as follows. To begin with, the number of flip-flops required is calculated for MOD-12 counter or Modulo 12 counter

$$\begin{aligned} n &= \log_2 N \\ &= \log_2 12 \\ &= 3.478 \end{aligned}$$

This value is rounded to the next highest whole number. The MOD-12 counter requires 4 JK negative edge triggered flip-flops. The binary equivalent for MOD-12 counter is

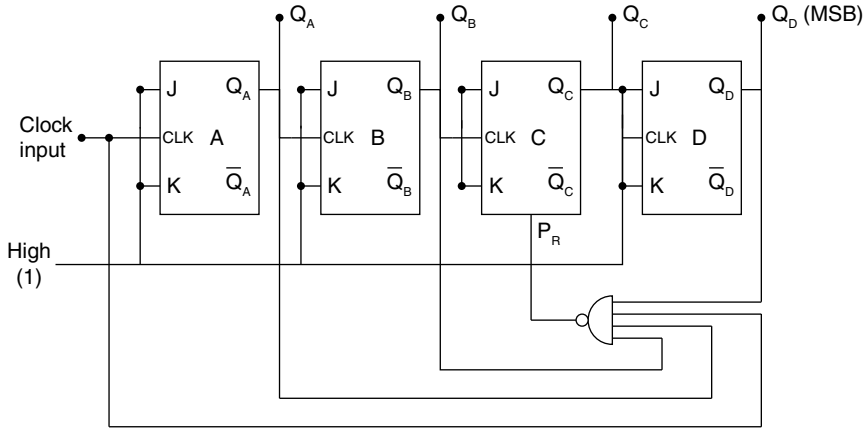
$$(n-1) = 12 - 1 = 11 = 1011$$

Thus, the output of the four FF_S are $1011 = Q_D Q_C Q_B Q_A$

It is evident that the outputs D, B and A are at level '1' and are connected as inputs to NAND gate. The output from the NAND gate is connected to the preset terminal of FFC since the output is 0. The J and K terminals of all FF's are connected to a logic

HIGH. The circuit is switched ON and works as a ripple counter. The counter has 12 states from 0 to 11. It keeps counting from 0 to 11 successively. On the arrival of the positive edge of the 12th pulse the presets are at 1 and the negative edge resets the counter to 0. Thus, the counter starts counting again from 0 to 11 till the 12th pulse is given. The output frequency of FFD is 1/12th of the clock frequency.

Thus, any MOD number is designed using this technique. The circuit and truth table is shown below.



Time	Pulse	Q_D	Q_C	Q_B	Q_A
t_0	0	0	0	0	0
t_1	1	0	0	0	1
t_2	2	0	0	1	0
t_3	3	0	0	1	1
t_4	4	0	1	0	0
t_5	5	0	1	0	1
t_6	6	0	1	1	0
t_7	7	0	1	1	1
t_8	8	1	0	0	0
t_9	9	1	0	0	1
t_{10}	10	1	0	1	0
t_{11}	11	1	0	1	1
t_{12}	12	0	0	0	0

Example 11.5 Design a synchronous MOD-5 counter, using Master/slave JK flip-flops to count states 000, 001, 011, 100, 110. The excitation table of JK flip-flops and K-map can be used for method of minimization.

Solution: The JK flip-flop are used and the number of flip-flops as calculated before is 3. The truth table and excitation table for JK flip-flop is shown below.

Truth Table

J	K	Q_{n+1}
0	0	Q
1	0	1
0	1	0
1	1	$\overline{Q}_n$

Excitation Table

Q_n	Q_{n+1}	J	K
0	0	0	ϕ
0	1	1	ϕ
1	0	ϕ	1
1	1	ϕ	0

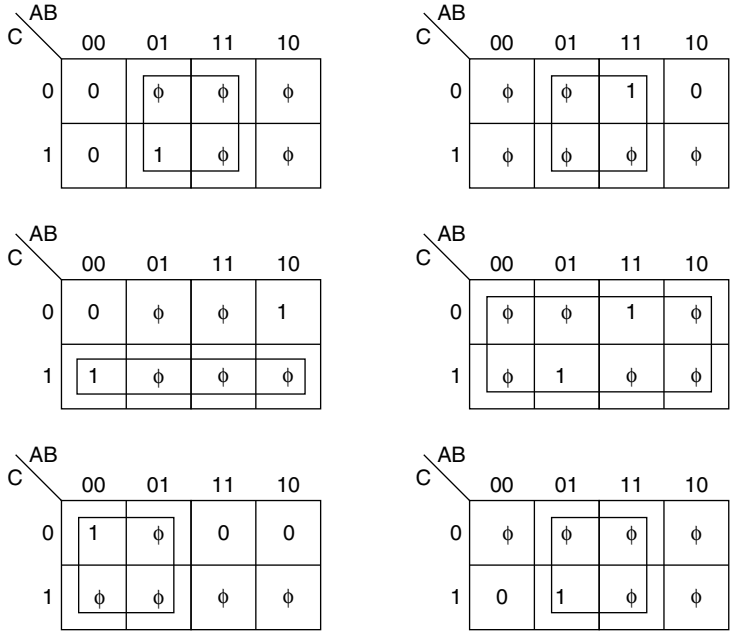
MOD-5 counter has the following states:

0 0 0
0 0 1
0 1 1
1 0 0
1 1 0

Using the excitation table the input of all the J and K terminals of the three flip-flops are interpreted. It is evident from the table that when the output changes from 0 to 0 then J takes 0 and K takes ϕ. When Q changes from 0 to 1, J takes 1 and K takes ϕ. Similarly, when the output changes from 1 to 0, then J becomes ϕ and K becomes 1. If Q changes from 1 to 1 then J takes P and K = 0. This is followed in considering the excitation table for MOD-5 counter which works only for the given states.

Time	Q_A	Q_B	Q_C	J_A	K_A	J_B	J_B	J_C	K_C
t_0	0	0	0	0	ϕ	0	ϕ	1	ϕ
t_1	0	0	1	0	ϕ	1	ϕ	ϕ	0
t_2	0	1	1	1	ϕ	ϕ	1	ϕ	1
t_3	1	0	0	ϕ	0	1	ϕ	0	ϕ
t_4	1	1	0	ϕ	1	ϕ	1	0	ϕ
t_5	0	0	0						

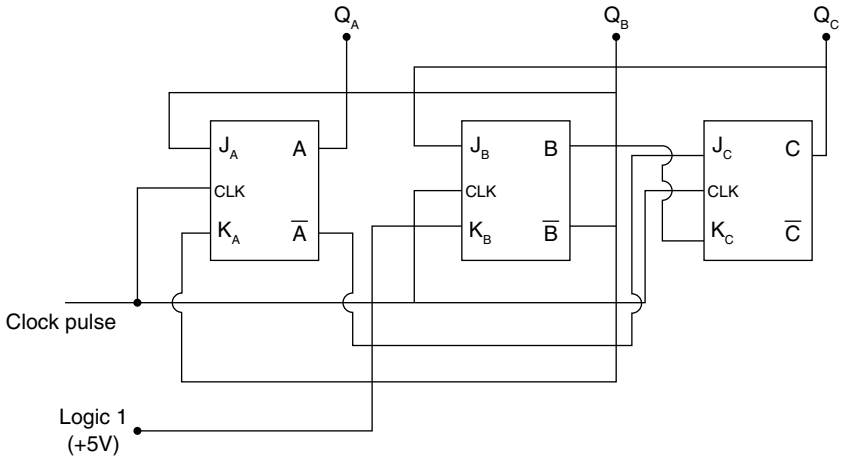
The Karnaugh maps for J and K inputs for all FFs are taken. A three variable K-map is required as it involves 3 variables A, B and C. The missing states other than the states mentioned are left empty in their respective cells. These cells can be assigning either 0 or 1 at the time of grouping. They also can be known as do not care states.



The values of J and K thus obtained are

$$J_A = \overline{B} \qquad K_A = \overline{B}$$
$$J_B = C \qquad K_B = 1$$
$$J_C = \overline{A} \qquad K_C = B$$

The design of the MOD-5 counter for the given states is



Examples 11.5 Design a synchronous MOD-5 counter using M/S JK flip-flop to run through states 000, 001, 011, 101 and 111 only. Use the excitation table of JK flip-flop to obtain design and also K-map method of minimization.

Solution: Three JK flip-flops are used and the excitation table and truth table are shown below.

Truth Table of JK flip-flop

J	K	Q_{n+1}
0	0	Q
1	0	1
0	1	0
1	1	$\bar{Q}$

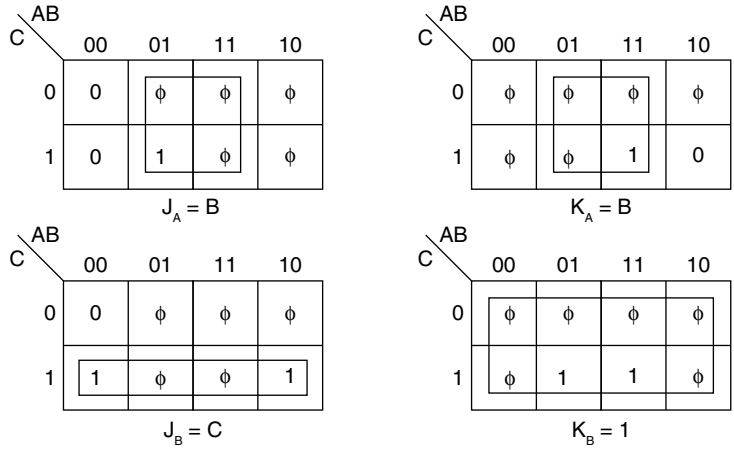
Excitation Table of JKflip-flop

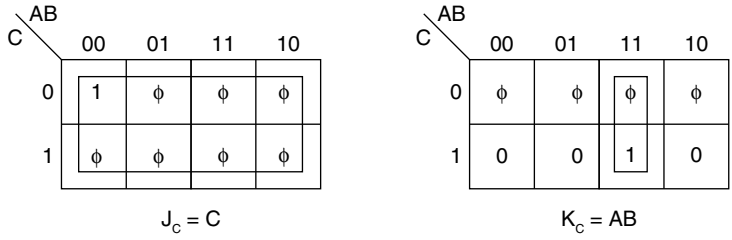
Q_n	Q_{n+1}	J	K
0	0	0	ϕ
0	1	1	ϕ
1	0	ϕ	1
1	1	ϕ	0

The states are 000, 001, 011, 101 and 111. The excitation table is constructed as before.

Time	Q_A	Q_B	Q_C	J_A	K_A	J_B	K_B	J_C	K_C
t_0	0	0	0	0	ϕ	0	ϕ	1	ϕ
t_1	0	0	1	0	ϕ	1	ϕ	ϕ	0
t_2	0	1	1	1	ϕ	ϕ	1	ϕ	0
t_3	1	0	1	ϕ	0	1	ϕ	ϕ	0
t_4	1	1	1	ϕ	1	ϕ	1	ϕ	1
t_5	0	0	0						

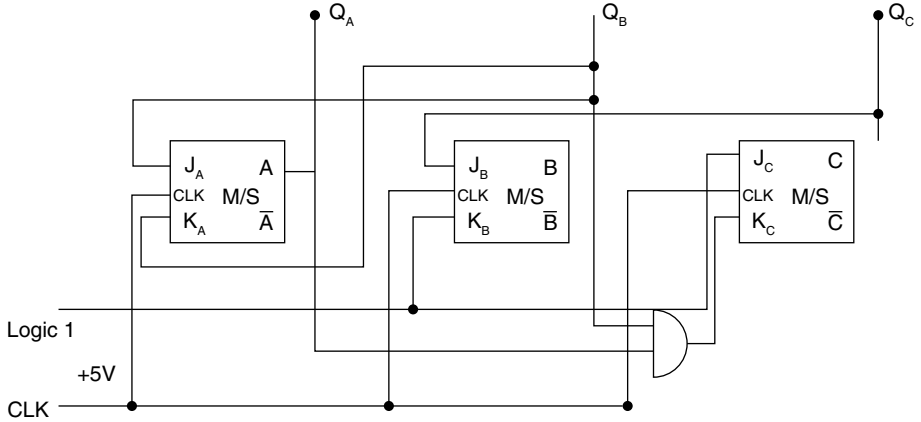
The K-map for the J's and K input of all the flip-flops are represented as





The following values are obtained by using K-map minimization

$$J_A = B \qquad K_A = B$$
$$J_B = C \qquad K_B = 1$$
$$J_C = 1 \qquad K_C = AB$$



MULTIPLE CHOICE QUESTIONS

1. To construct a modulus-6 ring counter _____ flip-flops are required.
   (a) 2 (b) 12
   (c) 6 (d) 3
2. A ripple counter of modulus-12 has clock frequency 30 KHz, then the output frequency of the counter is
   (a) 2.5 KHz. (b) 30 KHz.
   (c) 2.14 KHz. (d) 1 KHz.

3. An 'n'-bit ring counter has the _____ number of states in its counting sequence.
   (a) 2^n (b) $2n + 1$
   (c) $2^n - 1$ (d) n
4. Which of the following is the maximum binary number that can be counted by a ripple counter that has 4 flip-flops?
   (a) 0000_2 (b) 1011_2
   (c) 1111_2 (d) 0101_2
5. A Mod-10 counter is also known as
   (a) BCD decade counter (b) Twisted tail counter
   (c) Johnson counter (d) Ring counter
6. A 3-bit asynchronous binary counter has a propagation delay of 18ns in each of the three flip-flops. The total propagation delay is
   (a) 3ns (b) 6ns
   (c) 18ns (d) 54ns
7. A MOD-16 binary counter requires _____ flip-flops.
   (a) 2 (b) 4
   (c) 8 (d) 16
8. The last state of a MOD-9 binary counter is
   (a) 1111 (b) 0000
   (c) 0001 (d) 1001
9. When four MOD-6 counters are cascaded, then the overall modulus of the counter is
   (a) 4 (b) 216
   (c) 6 (d) 24
10. To construct a divide-by-32 counter, a counter with modulus _____ is required.
    (a) 8 (b) 16
    (c) 32 (d) 64
11. The delay problems in asynchronous counters are eliminated when the
    (a) Input clock pulses are applied simultaneously to all stages
    (b) Input clock pulses are applied to the alternative stages
    (c) Input clock pulses are applied only to the last stage
    (d) Input clock pulses are applied to first stage alone
12. When two or more counters are cascaded, then the resulting MOD number is equal to the _____ of their individual MOD numbers.
    (a) Sum (b) Difference
    (c) Product (d) Log

13. When a MOD-15 and a MOD-10 counter are cascaded and if input clock frequency is 60 MHz, what is the output frequency?
    (a) 500 kHz (b) 400 kHz
    (c) 1 kHz (d) 4 MHz
14. Which of the following is an invalid output state for an 8421 BCD counter?
    (a) 1111 (b) 0000
    (c) 1000 (d) 0001
15. A 2-bit asynchronous counter has _____ states.
    (a) 2 (b) 4
    (c) 8 (d) 16
16. A modulus-10 counter has the final output that occurs one time after every _____ clock pulses.
    (a) 10 (b) 20
    (c) 30 (d) 40
17. The lowest output frequency possible for a cascaded counter containing a modulus-3 counter, a modulus-4 counter, and a modulus-10 counter with 12 MHz clock frequency is _____.
    (a) 1MHz (b) 3MHz
    (c) 4MHz (d) 10MHz

Answers

1. (c) 2. (a) 3. (d) 4. (c) 5. (a) 6. (d) 7. (b) 8. (d) 9. (b)
10. (c) 11. (a) 12. (c) 13. (b) 14. (a) 15. (b) 16. (a) 17. (a)

DESCRIPTIVE QUESTIONS

1. Discuss in detail, the classification of counters.
2. Explain the working of a 3-bit ripple counter.
3. Mention the disadvantages of asynchronous ripple counters.
4. Explain the working of a 4-bit UP/DOWN counter.
5. Design a MOD-6 synchronous counter using J-K flip-flops.
6. Design a synchronous Modulus-5 counter using JK flip-flop to run through states 000, 001, 010, 011 and 101 only. Also design the excitation table and the Karnaugh map.

MEMORY

Memory devices are digital systems that store data either temporarily or for a long term. Digital computers to hard disks have built-in memory devices that can store the data of the user or manufacturer. The data can either be in the form of control programs or programs that boot the system. Hence, to store such huge amount of data the memory devices must have enormous capacity. The challenge is to build memory devices that have large capacity but cost effective. The memory devices must be capable of storing both permanent data and instantaneous data.

The memory devices are made of semiconductor materials. The basic capabilities of a memory is its ability to store a value; it should permit us to read the value that is saved and also alter the stored value.

Memory devices are categorized into many types. Based on the volatility, they are classified as volatile and non-volatile: volatile devices are those that lose the data once the power supply to the chip is switched off. Non-volatile are based on the access sequence. They are further categorized as random access time or throughout. If the memory is sequential, then the data can be accessed only in a particular sequence. On the basis of the speed of the memory devices, they are either access time or throughout. The devices can be accessed for a particular time period or they can be operated throughout the operation cycle. If the memory is in the random access form, any data can be retrieved in any order. Based on the access type, they are classified into

1. Random access memory (RAM)
2. Read only memory (ROM)

The memory devices can be combinational logic or sequential logic. The RAM is similar to register and is based on flip-flops but no clock pulses. On the other hand, ROM is a programmable logic. The high performances of modern computers are due to these high speed storage devices. The memory chips form the integral part of all devices having different customized features, apart from their primary function of storing and retrieving the data. Research in the memory chips over the last few decades has provided successful gadgets like the digital cameras and music players. Both types of memories are generally integrated circuits, which are small electronic circuits that consist mostly of semiconductor components. Memory devices are referred to as ICs, memory chips, or microchips. We shall first deal with RAM and later on, the ROM.

Random Access Memory (RAM)

Random access memory is an artefact of hardware evolution. As the name suggests, the data can be accessed in any order, which is in contrast to the restricted access provided by other memory systems. The access time is the same regardless of the type of data stored in. RAM is the main or primary storage device in computer systems. All programs and data are stored in RAM. A random access memory should be able to store many words but one per address. It should be capable of reading a word stored at a particular address. RAM also must be capable of rewriting the word that was saved at a particular address. RAM forms the main storage device when compared to hard devices. Therefore, modern operating systems run primarily in RAM and they load additional applications. RAMs are categorized as volatile and non-volatile. The block diagram of RAM is shown in Figure 12.1.

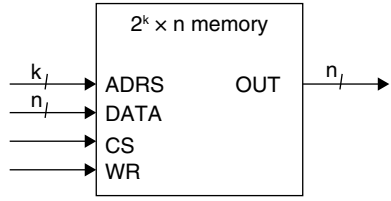


Fig.12.1 Block diagram of RAM

The above block diagram introduces the main interface to RAM. A chip select (CS) enables or disables the RAM. Address pin (ADRS) specifies the address or location to read from or write to. The read/write pin (WR) gives the option of reading or writing the selected data. When a data is to be read from memory, the pin WR must be set to logic 0. This allows the data stored at the address pin to be available at output. To facilitate the writing option, the WR pin is set to logic 1 which in turn allows a bit to be saved to the memory. This interface makes the RAM handy in applications where it needs to be combined to other RAMs to obtain a larger memory. There are two main control inputs on commercial memories: memory enable and read and write (read/write) which is shown in Table 12.1

Table 12.1 Memory enable Read/Write operations

Memory enable	Read/Write	Memory operation
0	X	None
1	0	Write to selected word
1	1	Read to selected word

The read/write operation has a few time constraints that are to be considered

- **Access time:** This is the time taken to read a word.
- **Cycle time:** This is the time required to write a word.
- **Set up and hold time:** The time taken by the address lines to hold the data.

Memory sizes are usually specified in numbers of bytes. One byte is equivalent to 8 bits. Therefore, a 2^{28}-bit memory translates into 2^{28} bits/8 bits per byte = 2^{25} bytes. This is equivalent to 32 megabytes. The size of RAM is measured in base 2 units, while hard drive size is measured in base 10 units. This is referred to as a $2k \times n$ memory. Where "k" refers to the address lines that specify one of $2k$ addresses. Each address contains an n-bit word. It has a storage capacity of $2k$ words. It has "n" data lines. These data lines are used for both input and output. The total memory size in terms of bits is given as

$$\text{Size} = 2k \times n \text{ bits}$$

For example, 1024-bit memory has 16-bit words. 10 bits are stored at each location as shown in Table 12.2.

Table 12.2 Memory address

Binary	Decimal	Memory content
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
	.	
	.	
	.	
	1021	100111010001111110
	1022	
	1023	

The main types of RAM include static RAM (SRAM), dynamic RAM (DRAM).

Static Random Access Memory (SRAM)

The static random access memory (SRAM) operates like a latch. Latches are preferred over flip-flops as the former compasses two NAND or two NOR gates, but a flip-flop requires at least twice the hardware of a Latch. Latches are smaller yet faster, cheaper, and consume less power. The SRAM, as the name defines is static, in the sense that they do not require constant electrical pulses to refresh but a constant current is applied to maintain the voltage differences. Static RAM is more expensive and has more capacity for storage than dynamic RAM that has to be refreshed more often, and is thus slower. Although SRAM requires less power, its power requirements keep varying with clock speed.

Each bit in a SRAM chip requires a cell implemented with six transistors. The storage density is thus less. The memory retains the information until the power is applied. The two primary applications of SRAM are in embedded systems and computer systems. It is used in automotive consumer electronics, industrial equipments, and for applications ranging from appliances to the small electronic toys.

Several megabytes of SRAM are employed in cell phones and music synthesizers. SRAM in computer systems perform small roles where high speed memory is required. Certain display units like LCD and networking devices such as routers, switches, and cable modems use SRAM. The static random access memory is different from the synchronous DRAM (SDRAM) or pseudo static RAM (PSRAM). A single SRAM cell is shown in Figure 12.2.

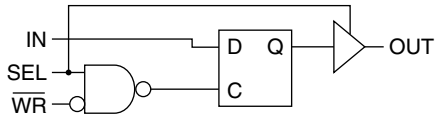


Fig.12.2 A single SRAM cell

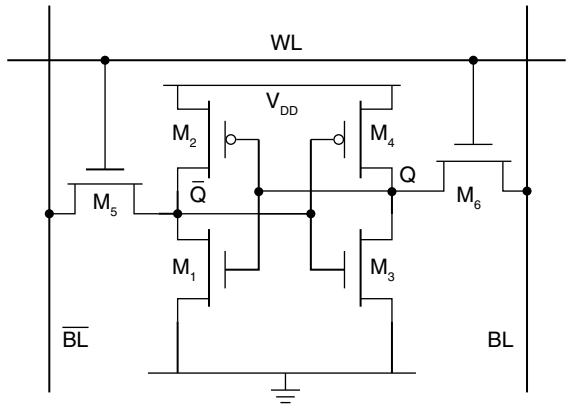


Fig.12.3 6-transistor CMOS SRAM cell

The 6-transistor CMOS SRAM cell consists of flip-flops M_1, M_2, M_3, M_4, M_5, and M_6 and is shown in Figure 12.3. The transistors M_1, M_2, M_3, and M_4 are all maintained at logic states by applying voltage to the gate terminals. The transistors M_1 and M_2 are in pairs when M_1 is ON then M_2 is OFF, this is the reverse in the next pair M_3 and M_4, i.e., M_3 is OFF and M_4 is in ON state. The cell is maintained at a particular state by the voltage given. During a read cycle, the two data lines float and when the word line goes high, the state of the cell is transmitted via the data lines to the sense amplifier. A 4×1 RAM is shown in Figure 12.4. Independent memory cells are used to make a 4×1 RAM. A four word requires two address bits. Each word is one bit therefore, the data input and output is also one bit each. The decoder attached to the CS inputs of the RAM cells selects the word. This type of RAM allows only one cell to be read or write at a time. The outputs are connected together with a single line.

A 4×4 RAM is shown in Figure 12.5. Each data input and output pins are now four bits long which enables to read and write four-bit words.

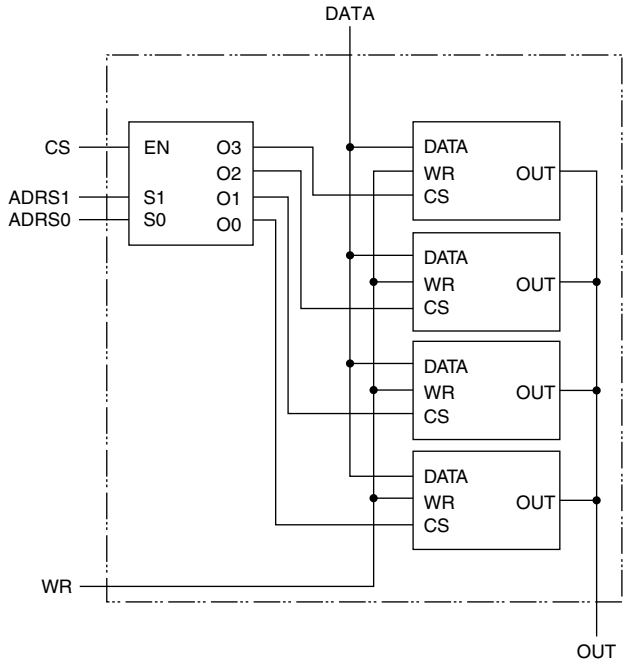


Fig.12.4 4 × 1 RAM

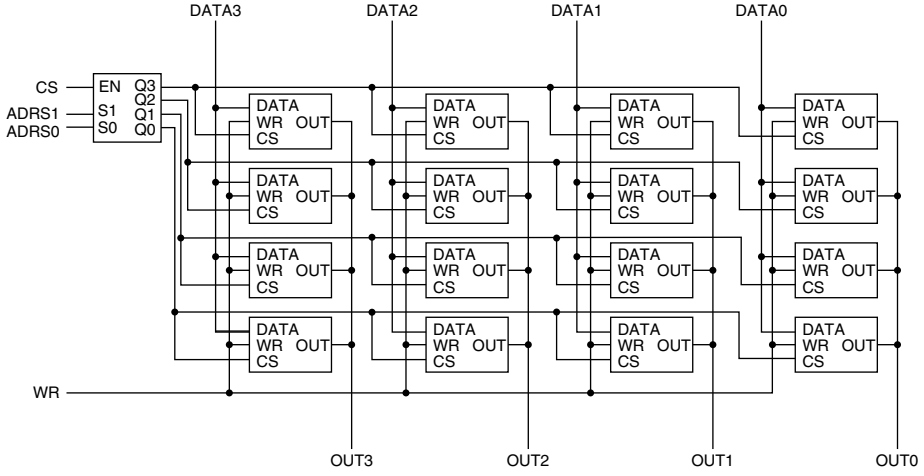


Fig.12.5 A 4 × 4 RAM

Consider an example of 64K × 8 RAM. 64K equates to 26 × 210 = 216, so there are 16 address lines and 8 data lines (Figure 12.6).

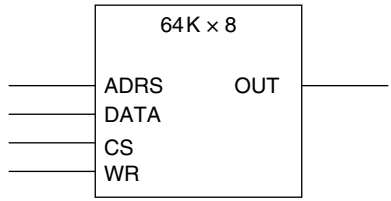


Fig.12.6 64K × 8 RAM

A 256K × 8 memory can be constructed from 64K × 8 chips as shown in Figure 12.7. 256K words would require 18 address lines. Two of the most significant address lines go to the decoder, which selects one of the four 64K × 8 RAM chips. The remaining 16 address lines are shared by the 64K × 8 chips. The 64K × 8 chips also share WR and DATA inputs. This assures that the 64K × 8 chips have three-state outputs. Thus, small RAMs are used as building blocks for making larger memories as in the previous examples.

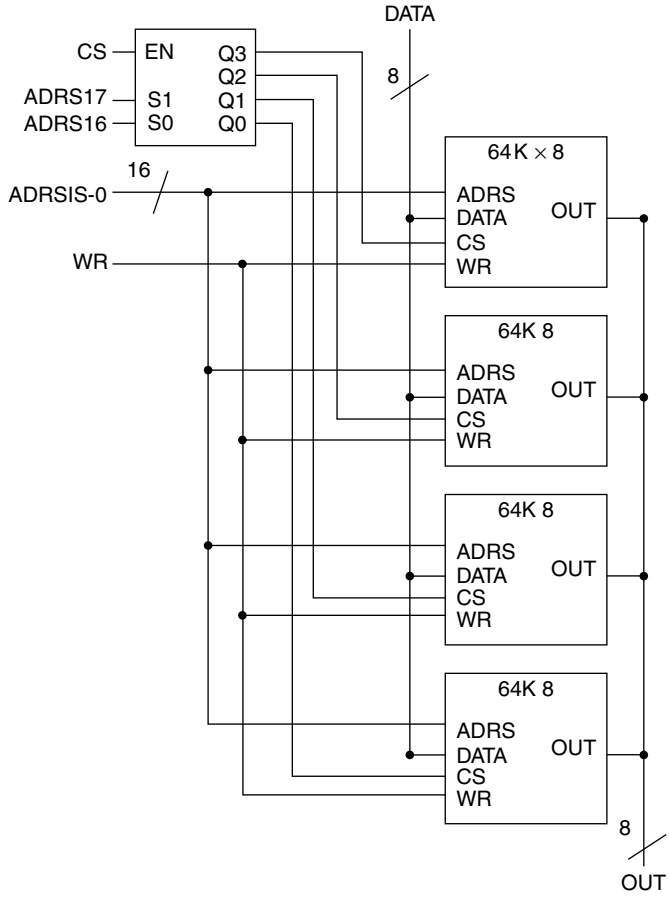


Fig.12.7 256K × 8 memory

DYNAMIC RANDOM ACCESS MEMORY (DRAM)

A single DRAM cell consists of one capacitor and one transistor memory cell. Storage of the charge on a capacitor is gated by a transistor. One transistor is employed for every bit. It is referred to as dynamic because the stored electrical state of the memory cells keeps changing continuously as the capacitor leaks the charge stored in it gradually. Thus, DRAM requires continuous refreshing so that the stored information is not lost. The memory system does this transparently. Memory cells of DRAM are shown in Figure 12.8.

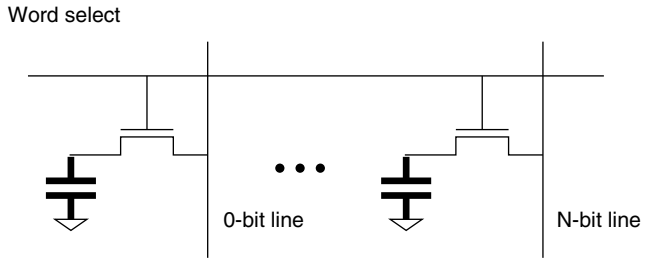


Fig.12.8 DRAM memory cell

Each storage cell capacitor must be refreshed every 60ns to 70ns approximately as defined by developing semiconductors standards. The DRAM has an inbuilt controller that performs the periodical refreshing. Thus, no software or hardware is required for refreshing the capacitors. Though it reduces the components, the circuit is yet complicated. This drawback is overcome by the cost of DRAM which is cheaper compared to the other memory types. DRAM has greater storage capacity and is much faster than SRAM thereby DRAM can be used as main memory. The refreshing is performed either row wise or all rows together in the limited time. Both the methods require a counter to note down the order of the row that has to be refreshed. This counter in most cases is inbuilt in DRAM.

The store and refresh operations are depicted as in Figure 12.9.

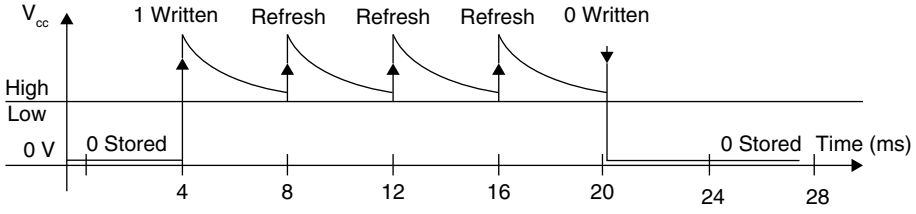


Fig.12.9 Store and refresh operations

The data can be written into the cell by taking the particular address line that is given logic high. This enables the transistor to "ON" state. The capacitor can be charged or discharged by the voltage on the data line. The capacitor charges for logic 1 and discharges for a logic 0.

The data on the particular address line can be read again when the transistor is turned on by making the cell on the address line HIGH. This makes the capacitor voltage (0 or 1) to appear on the data line. Every read operation is followed by a write operation as the data lost must be entered back to the cell. When the row and the column both are maintained at HIGH, then the data stored at the intersection of the row and column is either read or transferred to the cell for write operation.

External interferences, whether electrical or magnetic, inside the computer systems cause the bits to skip states. The interference can be of other forms caused due to background radiation that change the data stored in one or more memory cells directly or interfere in the circuit for READ/WRITE operation. This problem can be reduced by using more memory cells and memory controllers that would utilize these bits associated with DRAM. The surplus bits are used to check parity and to enable the data to be rebuilt by error correcting code (ECC). Single-bit errors can be checked using this parity detection. Double-bit errors can be detected, if the parity bits are increased.

The block diagram of DRAM is shown in Figure 12.10. It is clear from Figure 12.10 that DRAM is similar to SRAM with a few changes. The input pins keep increasing with the size of DRAM.

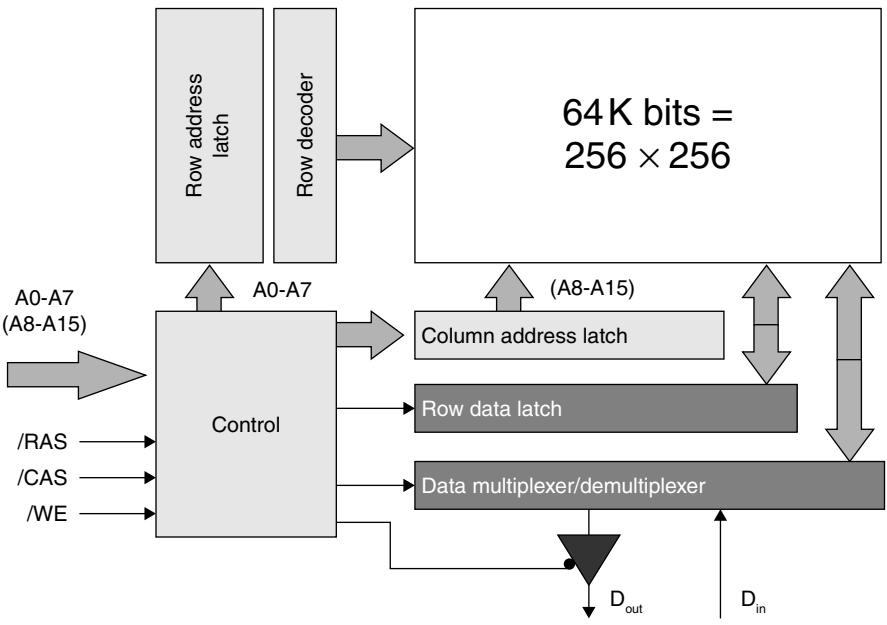


Fig.12.10 Block diagram of DRAM

A DRAM has few inputs and bidirectional data lines. There are four active-low control signals:

1. **Row address strobe (/RAS):** This pin enables to select a row during the trailing edge of /RAS. The row is held open as long as /RAS is slow.
2. **Column address strobe (/CAS):** The address inputs are captured on the falling edge of /CAS. It selects a column from the currently open row to read or write.
3. **Write enable (/WE):** If /CAS is high then it reads and when /CAS is low it writes. This pin determines read or write operation.
4. **Output enable (/OE):** This signal controls output to data I/O pins. The data pins are activated when /RAS and /CAS are low, /WE is high, and OE is low. Generally, /OE can be permanently connected low. While multiple memory chips are to be connected, this pin facilitates the parallel connections.

Although the RAM is asynchronous, the signals are typically generated by a clocked memory controller, which limits their timing to multiples of the controller's clock cycle.

GENERAL DRAM MODULES

Dynamic random access memory is manufactured as integrated circuits (ICs) connected and mounted into plastic packages with metal pins for connection to control signals and buses. Earlier these DRAMs were directly fitted on to the motherboard and later assembled into multi-chip plug-in modules (DIMMs, SIMMs, etc.). A picture of 256K × 4-bit 20-pin DIP DRAM is shown in Figure 12.11.

Fig.12.11 A 256K × 4-bit 20-pin DIP DRAM

DRAM chips are available in two formats. They are single in-pin package and dual in-line package.

Single In-line Pin Package (SIPP)

SIPP is a type of RAM that consists of small printed circuit board on which the memory chip is inserted. They have a single row of 30 pins that can be connected to the motherboard. These were used in the older systems and later were replaced by single in-line memory module (SIMMs) which proved to be user friendly.

Dual In-line Package (DIP)

These are electronic packages that are generally rectangular and have two parallel rows of connecting pins. These can be inserted to the circuit board or into a socket. These were discovered and developed in the 1960s to overcome the limitations of leads available in integrated chips.

DRAM MEMORY MODULES

Single In-line Memory Module (SIMM)

Single in-line memory module (SIMM) is a RAM memory module used in computers from the 1980s to 1990s. The SIMM differs from a dual in-line memory module (DIMM) by the arrangement of contact pins on the chip. Early computers used the dual in-line packages for their memory modules but these occupied more space on the motherboards. Use of more number of SIPP chips were minimized using a single enhanced memory module. SIMMs were found to be better alternatives. They were invented and patented by Wang Laboratories. They were originally built on ceramic and were provided with contact pins which were later removed and the modules were built on standard PCB material. The SIMMs had several variants, namely, 30 pins and provides 8 bits of data, the second variant of SIMMs has 72 pins and provides 32 bits. Based on the data bus width several combinations of the memory modules are installed either in pairs or identical pairs or groups of four.

Dual In-line Memory Module (DIMM)

It comprises of a series of dynamic random-access memory integrated circuits. They are placed on printed circuit boards and find their best use in personal computers, workstations, and servers. The SIMMs can be replaced by DIMMs due to the fact that the contact pins of DIMMs are present on both sides. The DIMMs have 64-bit data compared to the standard SIMMs that have a 32-bit data path. Many processors have a 64-bit bus width that requires identical pairs of SIMMs installed to occupy the data bus. This would require the processor to access the two SIMMs in parallel. This problem can be eliminated by introducing DIMMs. The most common types of DIMMs are 72-pin DIMM, 100-pin, 144-pin, 168-pin DIMM, 172-pin Micro DIMM, 184-pin DIMM, 200-pin SO-DIMM, 204-pin SO-DIMM, 214-pin Micro DIMM, 240-pin DIMM, 244-pin Mini DIMM. Small outline DIMM (SO-DIMM) are about half the size of regular DIMMs and are mostly used in notebooks, small footprint PCs, upgradable office printers, and networking hardware like routers.

Video RAM (VRAM) and Window DRAM (WRAM)

VRAM is a variant of DRAM and has dual in port. It is commonly used for storage of frame-buffer in some graphics adaptors.

WRAM is a variant of VRAM used in graphics adaptors. The performance of WRAM is better than VRAM and also less costly. It has greater bandwidth than VRAM.

Fast Page Mode DRAM (FPM DRAM)

The page mode helps in performing multiple read and write operations controlled by separate pulses. Thus, it enhances the performance speed while reading or writing the data.

Extended Data Out DRAM (EDO DRAM)

It is a minor modification to the fast page mode DRAM where new cycle can be started while the data of previous cycle is still active. This provides an extra performance speed.

Burst EDO DRAM (BEDO DRAM)

Burst EDO DRAM is an improvization of EDO DRAM that could process four memory addresses in one burst and hence saves clock pulses. This is done by adding an address counter on the chip that keeps note of the next address. The page access cycle is divided into two components. During the memory-read operation, the first component accesses the data that comes from the memory array to the output. The second component transfers the data bus from this latch to the appropriate logic level. Even though BEDO RAM was superior to SDRAM, still the latter technology quickly displaced BEDO due to the time it took to actually reach the market.

The other types of DRAM are multibank DRAM (MDRAM), synchronous graphics RAM (SGRAM), single data rate (SDRAM or SDR), and pseudo static RAM (PSRAM)

SDRAM (Synchronous DRAM) and DDR SDRAM (Double-Data Rate DRAM)

SDRAM is a type of DRAM that corresponds to the clock of the CPU's system bus.

DDR SDRAM (Double-Data Rate DRAM) is an improvization of SDRAM that allows data to be transferred on both the rising edge and trailing edge of a clock signal. This increases the amount of data that can be transferred in a period of time.

READ ONLY MEMORY

ROMs are integrated circuits that contain permanent data that cannot be altered. The data can only be read out of the memory and new data cannot be rewritten on to it. They are used wherever non-erasable data storage is required. They find their applications in microprocessors, controllers in cars, household appliances, etc. in computer systems they are used to store the "boot" or start-up program.

The ROM resembles the RAM organization. Memory can be stored at any desired location. This memory can later be driven out by addressing a particular location. The R/W pin is not required. The connections are done by diodes or transistors. Consider a ROM cell as shown in Figure 12.12. Every cell with data 1 is connected between a row and column. This is controlled by a voltage controlled switch. Cells with 0 are

not connected to row or column. By applying a voltage to the row, the required data is read and causes it to be displayed voltage at the column line.

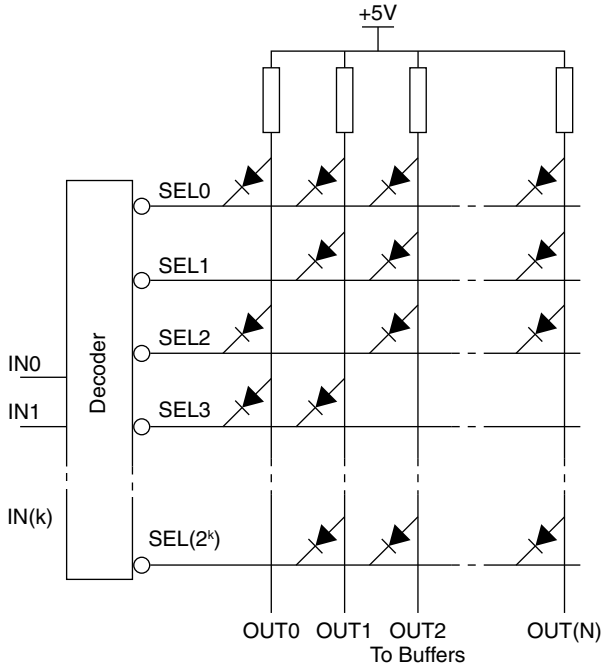


Fig.12.12 ROM cell

Diodes are used as voltage controller switch. ROM is programmed during the manufacture and cannot be changed. Since it is programmed during the manufacturing process, the data that has to be entered is to be mentioned by the user to the manufacturer. ROM can be implemented to perform a combinational logic function. Data stored in ROMs can be changed by supplying electrical pulses. These modifications in ROM led to the introduction of programmable ROM (PROM), erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), and flash, which is a type of EEPROM.

PROGRAMMABLE READ ONLY MEMORY

Programmable read only memory (PROM) is a modification of ROM that can be programmed according to the user's requirements. It allows the user to enter the data and programs. PROMs have special writing device attached to ROM that enables to enter the data. The PROM can be worked in two methods. The first method supplies all outputs of the PROM and is maintained at logic 0 and later the desired diodes are fused by applying logic 1. The second method is by applying all outputs by logic 1 and later

the desired transistors or diodes are blown out by giving logic 0. The working principle of PROM is simple. The data is addressed at a particular location and later a high voltage is given to the transistor. This fuses it and disrupts the connection between the rows and columns. It no longer connects the row and column lines at that location. The PROM can only be programmed once and cannot be reversed. These are used in the instruction storage components of home appliances.

EPROM AND EEPROM

EPROM is an acronym that stands for "Erasable Programmable Read Only Memory". The EPROM was the first reprogrammable ROM. In the EPROM, the logic 1 is stored by storage of charges and not by fuse blown method. The chips can be programmed and can store data for years. If the program is to be changed, then EPROM is exposed to high intensity UV radiation. The UV radiations pass through quartz window and later are directed to the chip. The contents of the chip are erased and the chip can be reprogrammed. The process can be completed in 20 minutes. EPROM are non-volatile, as they have the ability to store the data even when the power supply to the device is off. Initially these EPROMs are exposed to UV radiation of certain intensity which erases the chip. The new data is burned to the chip by the applied voltage and is ready to use. These chips are used to store the "boot" program of a computer, known as the firmware. They are also referred to as BIOS (Basic I/O System).

Electrically Erasable PROM (EEPROM)

This is a variation of the EPROM. Unlike the EPROMs, the EEPROMs are erased electrically and can be reprogrammed. A reverse polarity voltage is applied that removes the charge on the cell. The lifetime of EEPROM is less compared to its counterparts. The EEPROM restricts the number of times the data can be erased, whereas the updation can be done for any number of times. EEPROM is mainly used in devices like digital potentiometers, digital temperature sensors, and real-time clocks. EEPROM enables to store the calibration or similar data that is required even when the power is switched off or removed. EEPROM is more advantageous than EPROM due to the fact that the writing and reading time is much faster. EPROM has a few disadvantages like the reprogramming cannot be performed for particular cells but all locations can be erased completely. While reprogramming an EEPROM, it must be removed from the circuit as it might cause damage to the insulating layer of the chip. The major advantage of EEPROM is that it does not require any specific additional device.

FLASH MEMORY

These are special type of EEPROM that can be erased and reprogrammed. The EPROMs could erase only data only byte by byte. The disadvantage faced by EEPROMs were overcome by flash memory since here the chips can be reprogrammed

and rewritten about thousand times. The manufacturer gives voltage to a particular portion called blocks or to the entire cell and erases the data stored on the chip. Flash memory is a non-volatile storage device. It works faster than EEPROMs as it writes data in mass, generally 512 bytes in size instead of 1 byte at a time. Flash memory also helps in storing BIOS and sometimes referred to as flash BIOS. They are cheaper than EEPROMs. Flash memories are found in USB sticks, flash cards, and typically range in size from 1GB to 32GB.

Magneto Resistive RAM (MRAM)

The magneto-resistive RAM neither uses electric charge nor current flow to store the data but utilizes magnetic storage elements. Two ferromagnetic plates form the magnetic elements that can withstand a magnetic field. The plates are separated by a thin insulating layer. One of the plates is a permanent magnet set to a particular polarity while the other has a variable field that can be adjusted to match an external field to store memory. The device is constructed in the form of grid containing the memory cells. Reading of the data in the memory cell is achieved by measuring the electrical resistance of the cell. The cells are connected to transistors that allow the current to pass through the cell. When a particular cell is selected by applying current to its associated transistor, then due to magnetic tunnel effect the cell enables the magnetic plates to change their orientation. This change in orientation causes their resistance to change. Measurement of the resulting current can determine the polarity. If both the plates have the same polarity this is considered to mean "1" and when both plates are of opposite polarity then resistance will be higher and this means "0". MRAM are similar to DRAM. MRAM never requires a refresh. MRAM has lower power consumption as they erase data block wise. Yet MRAMs are not preferred over DRAM and SRAM commercially.

PROGRAMMABLE ARRAY LOGIC

The programmable logic is similar to PROM. It consists of a network of AND-OR gates arranged to form a matrix. The matrix consists of AND array and OR array. The OR array of PAL is fixed while the AND array is programmable. It has a combined function of programmable logic array and PROM. The PAL has a number of inputs that can be AND-ed together across a given plane and the output is generated. The AND output is available for all possible combinations. The inputs to AND gate can have both input and its complement. The sum of the AND outputs can be obtained by the OR array. It is to be noted that the PLA has specific OR connections and all possible AND combinations. Thus the PAL can be used to implement Boolean expressions in the sum of the products form. This is performed by connecting or disconnection the connections between the gates. AND-OR gates are in real NAND gates. The fuses are represented by a cross (X). The block diagram of PAL is as shown in Figure 12.13.

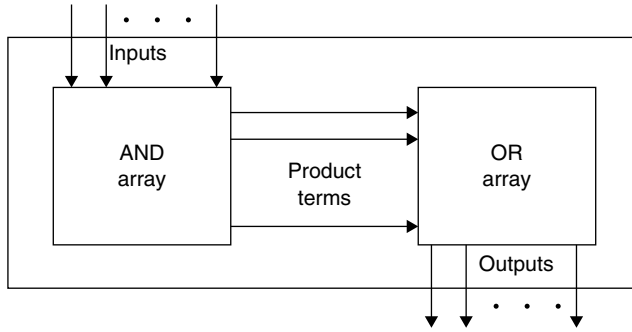


Fig.12.13 Programmable array block diagram for sum of products form

Unwanted connections are "blown" using the fuse. The fuse is normally connected and breaks the unwanted connections while the anti-fuses are generally disconnected and are used in making desired connections. The labeling of PAL chip has the number of inputs and outputs indicated on it. For e.g. 10H8 has 10 inputs and 8 active high outputs, 12L6 has 12 inputs and 6 active low outputs.

Consier the example depicted in Figure 12.14. The intersection of a row and column contains a fuse. When there exist a cross mark it symbolizes a intact connections and the nonappearance of a cross signifies a blown out connection. The output of the AND gates result in product terms which are fed to the inputs of the OR gate. Thus the output from the diagram is interpreted as

$$F_0 = A + B'C$$
$$F_1 = AB + AC'$$
$$F_2 = B'C + AB$$
$$F_0 = A + BC$$

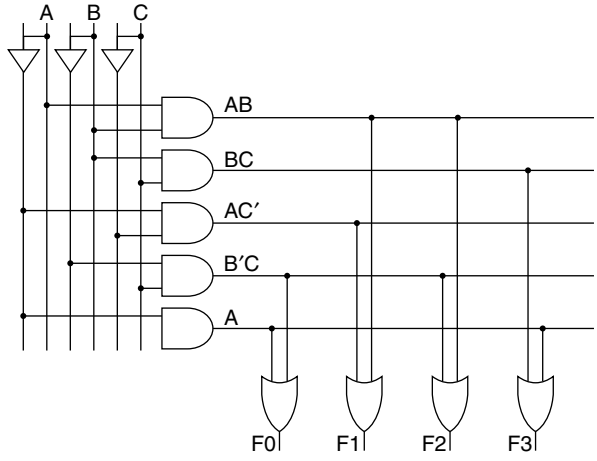
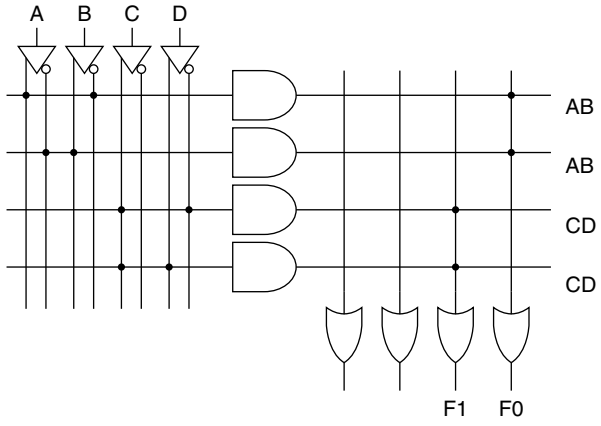


Fig.12.14 PAL architecture

Example 2.1 Implement the logic function $F_0 = AB' + A'B$, $F_1 = CD' + CD$ using PAL.

Solution:



PROGRAMMABLE LOGIC ARRAY

The programmable logic array has AND gate array and OR gate array. Unlike the PAL, the OR and AND gates both are programmable. The basic architecture of PLA is shown in Figure 12.15. The PLAs are more flexible than PALs, but PALs operate faster, because hard-wired connections take less time to switch than their programmable equivalents. The desired function is performed by blowing the fuses at specific points. They are also quite versatile, since both the AND and OR terms can have many inputs.

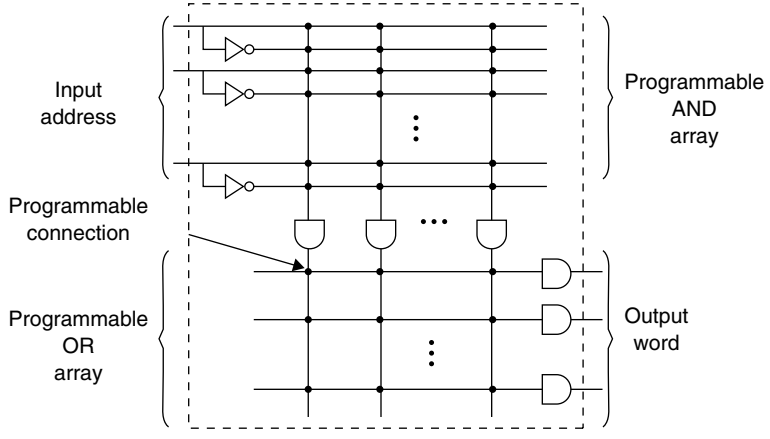


Fig.12.15 PLA structure

The PLA has design tools for multi-output minimization and it involves relatively lesser minterm combinations. The minterms can be combined at the output functions. The PLA also has few limitations. The design is more complex and thus can implement any function in the POS form. Since it has two programmable planes it operates much slower.

Difference Between RAM and ROM Memory

Table 12.3 Differences between RAM and ROM memory

Definition	Random access memory or RAM is a type of memory where the data stored can be accessed in any order from any desired location.	Read only memory or ROM is a type of memory where the data is stored permanently.
Use	It is used for quick performance of computers. It permits reading and writing of data.	ROM stores program that boots the computer. It allows only reading of the data.
Volatility	All stored data are lost once the RAM is switched off.	ROM is non-volatile. The contents are retained even when the device is powered off.
Types	The two main types of RAM are static RAM (SRAM) and dynamic RAM (DRAM).	The types of ROM include programmable ROM (PROM), erasable programmable (EPROM), and electrically erasable programmable (EEPROM).

SUMMARY

High speed and storage are two main requirements for all memory elements of electronic devices. RAM find its application in the main memory of computers. Dynamic RAM and static RAM are the major types of RAM. The Dynamic RAM (DRAM) requires constant refreshes of electrical current while the Static RAM consumes less power and is faster than DRAM. Owing to the fact that the DRAMs are less expensive, it is the most widely used type of solid-state memory. The static and dynamic RAMs are further classified into the other types of RAM.

Read Only Memory (ROM), are memory devices that perform only the read function and cannot be used for WRITE operation. Some modifications in ROM led to the programmable ROM (PROM) that can be programmed and used for desired functions. The development of electrically erasable programmable ROM (EEPROM) replaced the erasable programmable ROM (PROM). EEPROM is still used to store BIOS code on computers. Flash memory is commercially successful variation of EEPROM that can be used like RAM.

MULTIPLE CHOICE QUESTIONS

1. One byte is equivalent to
   (a) 1 bit (b) 10K bit
   (c) 8 bits (d) 7 bits
2. ROM using bipolar transistors has an access time of
   (a) 1 msec (b) 1 μsec
   (c) 1 nsec (d) 1 sec
3. The memory that requires refreshing is
   (a) SRAM (b) DRAM
   (c) ROM (d) EPROM
4. Which of the following is not accessed randomly?
   (a) DRAM (b) SRAM
   (c) ROM (d) Magnetic tape
5. Magnetic tape is accessed
   (a) Sequentially (b) Randomly
   (c) Both (a) and (b) (d) None of the above
6. The time taken to read a word is
   (a) Hold time (b) Cycle time
   (c) Access time (d) None of the above
7. How many address bits are required to represent 8K memory?
   (a) 13 bits (b) 12 bits
   (c) 8 bits (d) 10 bits
8. Memory can be classified on the basis of the _____ technology used.
   (a) Size (b) Bipolar
   (c) Fabrication (d) None of the above
9. To increase the speed in RAM, the bit storage cells are made of
   (a) BJT (b) NMOS
   (c) Diodes (d) Transistor
10. Which of the following memories stores the most number of bits?
    (a) 64K × 8 memory (b) 6M × 8 memory
    (c) 64K × 6 memory (d) 32M × 8 memory
11. ROMs are integrated circuits that contain
    (a) Temporary data (b) Permanent data
    (c) Programmable data (d) None of the above
12. Read Only Memory (ROM) can be used for _____ information from it.
    (a) Read (b) Write
    (c) Erase (d) Both Read and Write

13. The memory cell of dynamic RAM consists of
    (a) A capacitor (b) A transistor
    (c) A flip-flop (d) A transistor acting as a capacitor
14. The data is stored in Magneto-resistive RAM through
    (a) Electric charge (b) Current flow
    (c) Magnetic storage elements (d) None of the above
15. RAM stands for
    (a) Random origin memory (b) Random only memory
    (c) Read only memory (d) Random access memory
16. The data in EPROM can be erased with
    (a) Fluorescent (b) UV rays
    (c) LED (d) Laser
17. The input pins of DRAM _____ with the size.
    (a) Decreases (b) Doubles
    (c) Increases (d) Remains same

Answers

1. (c) 2. (b) 3. (b) 4. (d) 5. (a) 6. (c) 7. (a) 8. (c) 9. (a)
10. (d) 11. (b) 12. (a) 13. (d) 14. (c) 15. (d) 16. (b) 17. (c)

DESCRIPTIVE QUESTIONS

1. Compare the memory devices RAM and ROM.
2. How does a static RAM cell differ from a dynamic RAM cell in its working?
3. What are the main drawbacks of dynamic RAM compared to a static RAM?
4. List the advantages of dynamic RAM compared with static RAM.
5. Discuss in detail the various types of ROM.
6. Distinguish between a PROM and an EPROM. What are their disadvantages?
7. Mention the applications of ROM.
8. What are programmable logic arrays? How is it different from programmable array logics.
9. Write short notes on
    a. Magneto Resistive RAM (MRAM)
    b. EEPROM
    c. Flash memory
    d. Cache memory

DISPLAY DEVICES

Display units play a vital component of electronic devices. They are used to exhibit the information. Display devices find a wide range of applications from digital watches to billboard displays. There are many types of display devices available, a few of which will be discussed in this chapter, namely, CRTs, plasma displays, LCD, LED, 7-segment displays, OLED and FOLED, digital light processing, and e-books.

Display technology is generally classified as projective displays and reflective displays. Further projective displays are divided into two classes: emissive and transmitive. CRT and gas plasma displays fall under emissive type of technique, whereas liquid crystal displays (LCD) and liquid crystals on silicon (LCOS) are transmitive types of display technology. On the other hand, the digital light processing (DLP), organic led displays (OLED), and e-books are reflective displays.

CATHODE RAY TUBES (CRTs)

Cathode ray tubes are primitive display devices being extensively used for ages. They are commonly known as the picture tubes by people as their major application was found in television sets. Since vacuum tubes form an integral component of the CRT, the clear understanding of the working of VT is essential. Vacuum tubes (VT) were employed as valves and switches before the invention of transistors, electronic chips, and integrated circuits.

It has four active elements encapsulated in an evacuated tube. The voltage supply to the heater coil, the voltage across cathode and plate are extended through wires along the base of the tube. The grid is supplied a control voltage. The device works as a valve or switch. A small change in control voltage produces a large change in current across the cathode circuit. The small control voltage regulates the valve, i.e., it can turn the current to ON or OFF state. The cathode ray tube is shown in Figure 13.1.

Working

The cathode being negatively charged serves as a reservoir of electrons. The coil is heated by the voltage applied which in turn heats the cathode thereby imparting energy that causes release of excess electrons. These free electrons are attracted to anode, a positively charged plate. The flow of electrons from cathode to anode results in current flow across the circuit. The grid is closer to the cathode than the anode. The voltage

at grid determines the acceleration of electrons towards the plate; if the grid is more positive it attracts more electrons and increases the speed of electrons towards the plate. If the grid is maintained at a negative potential, then the flow of electrons to the plate is inhibited. The flow of electrons from the cathode is due to secondary electron emission.

The working of a CRT is similar to a simple vacumm tube with few changes in the internal components of the circuit. The flow of electrons from cathode is similar to that of a vacumm tube. In a CRT, once the electrons reach the grid, they are focused on to a glass screen that acts as a plate. The focusing coil aligns the electrons into a narrow beam just as a lens converges ray of light. These steering coils are capable of turning the beam either to left or right, up or down thereby directing it to a particular spot on the screen. When the electrons fall on the phosphor coated screen it glows. The intensity or brightness of the glowing phosphor dots depends upon the intensity of the electrons, which is determined by the grid voltage.

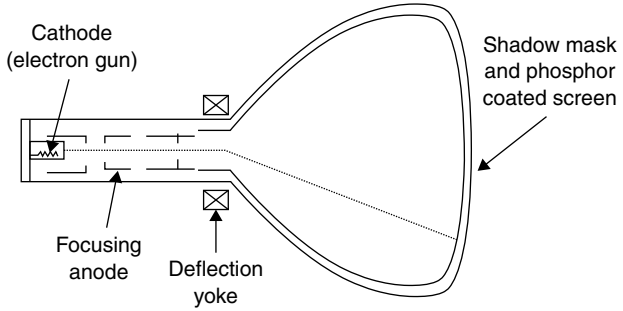


Fig.13.1 Cathode ray tube

A CRT can be used to display a picture in two ways.

1. The electron beam can be directed to draw a line on the screen. The line is drawn repeatedly at high speed that gives an illusion of permanent image. This method is similar to etching. This type of display is known as vector display, and is popular in computer graphic and computer aided design (CAD).
2. The second type of technique is raster technique. This works by scanning the electron beam over the screen in regular pattern of scan lines to paint out a picture. Whenever a scan line is identified across the screen by the beam, the beam is modulated. The modulation of the beam is proportional to the brightness of the point on the picture. Once the scan line is drawn, the beam is turned off before moving to the starting point of the next scan line. The resulting pattern is called raster and is commonly known as bitmap.

Colour CRTs

Colour CRTs works similar to monochrome CRT but the tube has three separately controlled electron guns that produce the electrons. The screen has phosphor dots of primary colour, namely, red, green and blue. Each of the three electron beam is calibrated to illuminate only one phosphor colour. Therefore, though the beam of electrons

do not have colour since the electrons go and activate the phosphor dots they can be considered of having red, green and blue electrons. Colour screens are usually made up of RGB system and optical mixing of adjacent dots take place in the eye. It has a typical triangular or triad arrangement of the coloured phosphor dots on the back of the glass screen of colour CRT screens and is shown in Figure 13.2. An opaque shadow mask is placed between the three electron guns and phosphor to ensure that each gun excites only phosphor dots of appropriate colour.

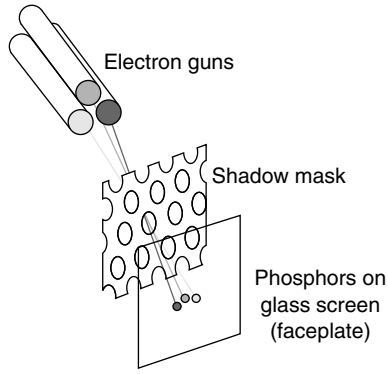


Fig.13.2 Colour CRT

Frame Buffers

Frame buffer is an arrangement of computer memory to store colour information similar to an electron gun as in CRT. Each row of the array fetches 1 pixel value (0 or 1). This pixel value is used on the electron gun which turns it ON for white and turns it OFF for black.

The grey scale frame buffer uses a digital analog convertor to convert numeric grey level to different voltages. Each pixel can take a value of 0–256 for each (R, G, B) thereby focusing palette of 256 colour. But colours are drawn from a selection of nearly 17 million 16,777, 216. A complete colour resolution frame buffer is known as true colour frame buffer. High display has more than 24 bits/pixel to handle colour compositing for real-time animated displays and overlays. The information from frame buffer to CRT is considered linear. Any linearity introduced by human perception and the electronics of CRT is done by gamma corrections. The actual phosphor intensity I_a due to grid voltage V is given as:

$$I_a = I_\gamma, \text{ where } \gamma \text{ is time constant y than l}$$

The value varies between 1.6 and 2.4 neglecting the human perception error; γ correction consists of each pixel value from $0^{1/\gamma}..1^{1/\gamma}$….etc.

Human eye does not respond linearly to light but rather responds to relative brightness or luminance difference as stated by Weber's law. To overcome this, the CRTs have inherent gamma correction which gives a uniform perceptual coding.

Gamma describes the non-linear relationship between pixel values and luminance.

$$\text{Value out} = \text{value } I_n^{\gamma}$$

If $\gamma < 1$ corresponds to gamma encoding and $\gamma > 1$ corresponds to gamma decoding. This results in a gamma correction for display given by the relation.

$$\text{Luminance} = c^* \text{value}^* + \text{black level}$$

Where, c = contrast control set by the monitor,
value = pixel level max of 1 and
black level = the brightness level control set.

LIQUID CRYSTAL DISPLAYS (LCD)

Liquid crystal display devices are categorized under flat panel displays. Liquid crystals were discovered in 1888 by an Austrian botanical physiologist Friedrich Reinitzer. These crystals are complex organic molecules that are arranged in a rigid lattice.

He observed that cholesterol benzoate had two transition states, viz., a cloudy liquid and other transparent state. Matter in general is exhibited in three states, solid, liquid, and gas. Solid states can be further classified into crystalline and amorphous based on the arrangement of molecules. Crystalline solids have a regular arrangement of molecules while amorphous solid has an irregular structure. When the crystalline solids are heated they become *isotropic liquids* and are known as crystalline liquid or liquid crystals. They possess an intermediate state of matter between liquids and solids and have the fluid characteristics of a liquid but also possess the molecular orientation of a solid. These are long cigar-shaped molecules that move like any liquid molecules, but in addition these molecules are held together in a lattice. They exhibit electric, magnetic, and optical anisotropy.

Liquid crystals are classified as smectic, nematic, and cholesteric based on their optical configuration. Figure 13.3 represents the orientation of the globules inside each type of liquid crystal. The nematic type of liquid crystal is of our concern as they are the most commonly used crystal for displays. The twisted nematic liquid crystal is a special type where the crystals are numerically twisted through a certain angle by applying voltage. When current is applied to these crystals, they untwist. The amount of untwisting is directly related to the strength of the current applied.

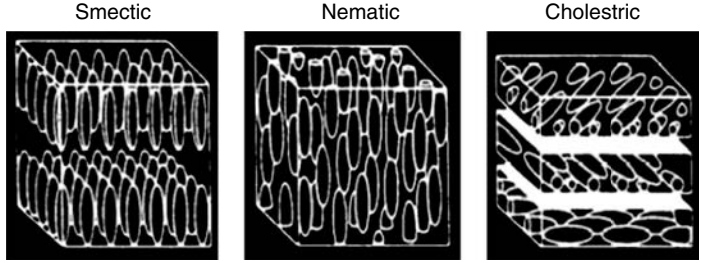


Fig.13.3 Orientation of liquid crystals

LCD's are used as light directors and not producers. They do not produce their own light like other display devices. They get light from an external source which is later modified by them. The external sources can be built inside the device, as they

only modify the light and do not produce. The power consumption of a liquid crystal display is less since the light source can be surrounding light or an inbuilt light source. The LCD receives light from this source which is placed at the back portion of the device. Therefore, it is also referred as back-lit LCD.

To understand the working of a liquid display, one has to primarily know the principle of polarizer. Light comprises of tiny light particles known as photons. Light waves travel as transverse waves which consist of both electric and magnetic fields perpendicular to each other and to the direction of propagation. The orientations between these fields give rise to different polarization. Unpolarized light thus has multi vibrations and it can be converted into polarized light by passing them through a polarizer or filter. A polarizer blocks all directions of light vibrations and allows light of single vibrational state to pass through them.

The polarizing filter has a number of small parallel lines as depicted in Figure 13.4. These lines are constructed on a molecular level by the transparent chemical compounds that make up the filter. The polarizing filter will allow only the light waves parallel to the filter's lines to pass through it. If the light wave is perpendicular to the slits on the polarizer, it is blocked by either reflection or absorption. When light wave is to be passed through two individual polarizers that are crossed with each other then both must be placed parallel to each other and to the direction of the incident light to receive the polarized light wave. This is explained below.

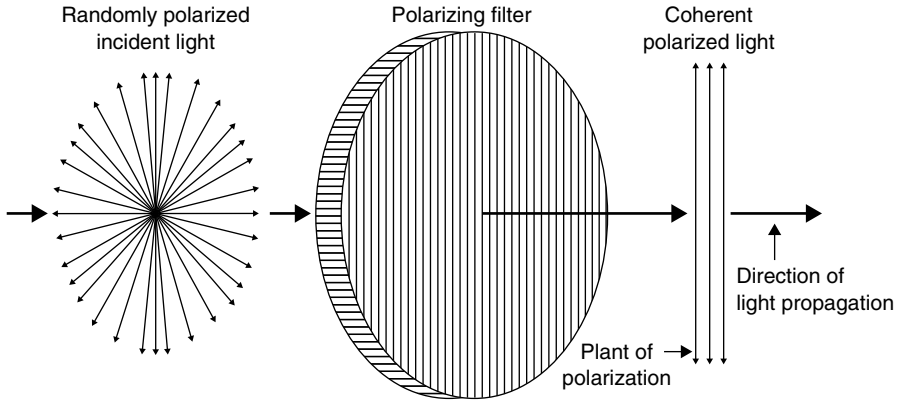


Fig.13.4 Polarizing filter

Consider two polarizers placed one beside the other. When an unpolarized light passes through the first polarizer it is converted to linearly polarized light. It then passes on to the second polarizer which is also placed in the parallel direction. This allows the light to be polarized and finally light emerges out.

Consider a second possibility when first polarizer is placed parallel to the incident light and the second is placed perpendicular to the direction of the propagation. Light is linearly polarized by the first filter and is passed onto the second polarizer. Since the direction of propagation of the light falling is perpendicular to the second filter, the light is blocked. Light is either absorbed or reflected by the filter and no light emerges

out. Figure 13.5 shows the working of polarizing filter. Thus, two polarizing filters can be used together to stop the transmission of light. The working of a twisted nematic liquid is similar to the polarizer and is explained below.

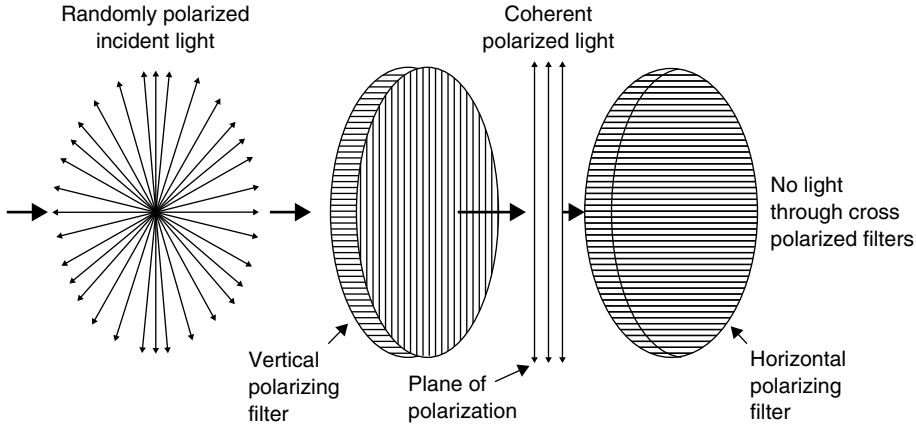


Fig.13.5 Working of polarizing filter

Types of LCDs

LCDs are of many types, namely

- **Dynamic scattering:** They require high-operating voltage and higher power consumption. The display is less legible and this type of display is outdated.
- **FLC (Ferroelectric Liquid Crystal):** They have faster switching times and better grey scale.
- **TN:** Twisted Nematic Liquid Crystal
- **STN:** Super-twisted Nematic
- **TFT:** Thin Film Transistor Active Matrix TN

We shall restrict our discussion only to the working of the twisted nematic liquid crystal.

Construction of LCD Display

The LCD is constructed by placing a twisted nematic liquid crystal display between the upper and lower substrate plates. These plates act as the electrodes that conduct the electric current passed through the liquid crystal. The two substrate plates are separated by a distance of 6–8 μm. The substrate plates are made of transparent conductive coatings (transparent electrodes) of indium-tin oxide (ITO) on their inner surfaces. The electrodes are transparent and patterned on the glass by photolithography. The electrodes are placed to form a grid over the display. Each cell has two electrodes, one which is particularly of the cell and another shared in common with all the other cells. The common electrode is often called the "backplane" of the LCD. The construction is shown in Figure 13.6.

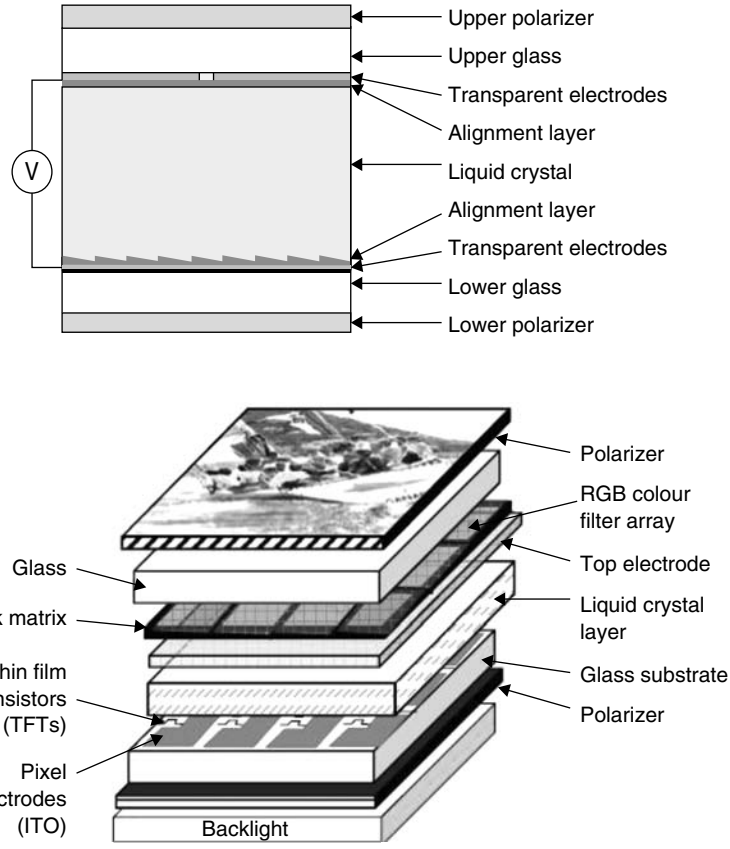


Fig.13.6 Construction of LCD display

Working of LCD with a Twisted Nematic Crystal

The LCD uses an arrangement of filters to display information. LCD has a twisted nematic crystal sandwiched between two polarizers. The surrounding light falls on the front polarizing filter. The light is linearly polarized and then passes through the liquid crystal medium. This nematic crystal medium has the capacity to rotate the light passing through them. They rotate light from a few degrees to over 270°. When the voltage supplied to the crystal through the electrodes is in "OFF" state, the light is continuously twisted by 90°. When a desired voltage is applied to the electrodes the optic axis of the liquid crystal orients to the direction of propagation of light and the twisted structure disappears. Now the voltage is in "ON" mode. The direction of the polarized light is no longer rotated and light passing through the cell intersects the second polarizer in the crossed position where it is absorbed, causing the activated portion of the display to appear dark. The working of LCD with twisted nematic crystal is shown in Figure 13.7.

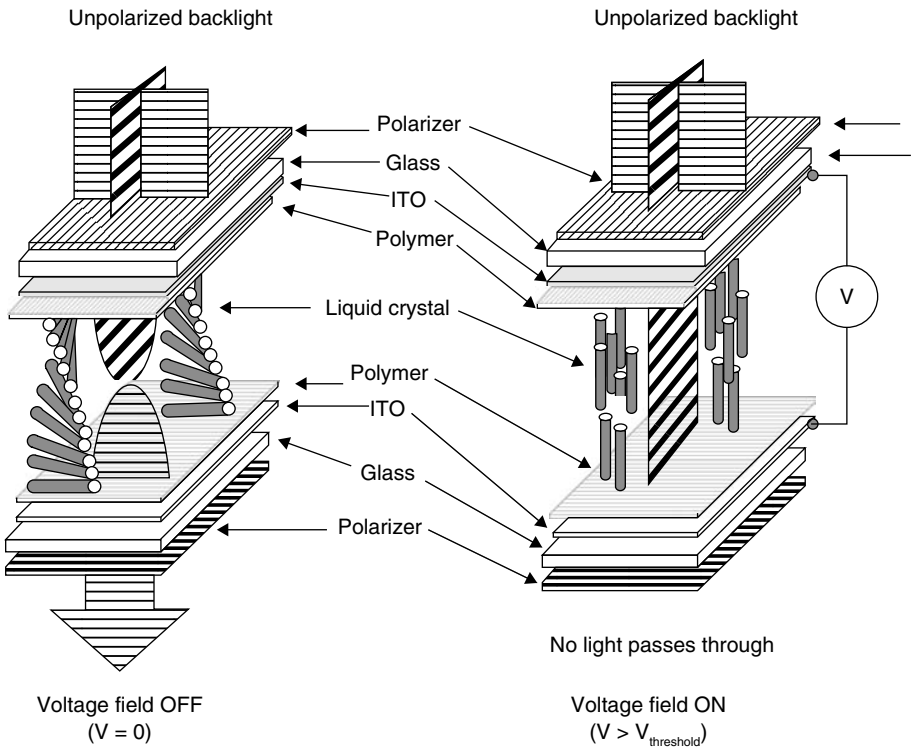


Fig.13.7 Working of LCD from twisted nematic crystal

Colour LCD

In colour LCDs, the pixels act as light valves. The pixels control the amount of light passing through the red, green, and blue mosaic filters. The colour rendition is an important aspect for any display; therefore, the colour balance of three colours is to be balanced throughout the display. This can be achieved by using a monochromatic light but in actual other wavelengths leak to produce colour. The thickness of typical red, green, and blue mosaic filters is chosen such that the cell gap under the filters is 6.4 μm, 5.8 μm, and 4.8 μm, respectively. The multi gap design not only ensures a true black but also a good colour balance over the whole dynamic range of the display. The pixels are controlled in an LCD display and shown in Figure 13.8.

Super Twisted Nematic LC: When chiral nematic liquid crystals are doped with optically active materials they form super twisted nematic (STN) liquid crystals. In high-definition LC displays, contrast and viewing angle is a drawback because of the low on/off voltage. This can be overcome by making the TN crystals to twist light for angles greater than 90°. To maintain a twist angle greater than 90° requires a nematic liquid crystal having a twisted structure known as a chiral nematic.

The advantage of STN displays is that they have a much wider range of viewing angles and contrast.

- **Passive matrix:** It is a type of control system where a simple grid supplies the charge to a particular pixel on the display. It has a slow response time and imprecise voltage control.
- **Active matrix:** In this type of operating pattern, every pixel has switch and capacitor. A row is switched on, and then a charge is sent down a column. Capacitor holds charge till next cycle. They have faster response time and less pixel crosstalk. An enormous number of transistors are used. For example, in a laptop, 1,024 $\times$ 768 $\times$ 3 = 2,359,296 transistors are etched onto the glass.

The control circuits can be driven by two ways, namely, direct drive method where every element is wired separately and multiplex driving where the elements are interconnected like a matrix. The various control circuits are depicted in Figure 13.8.

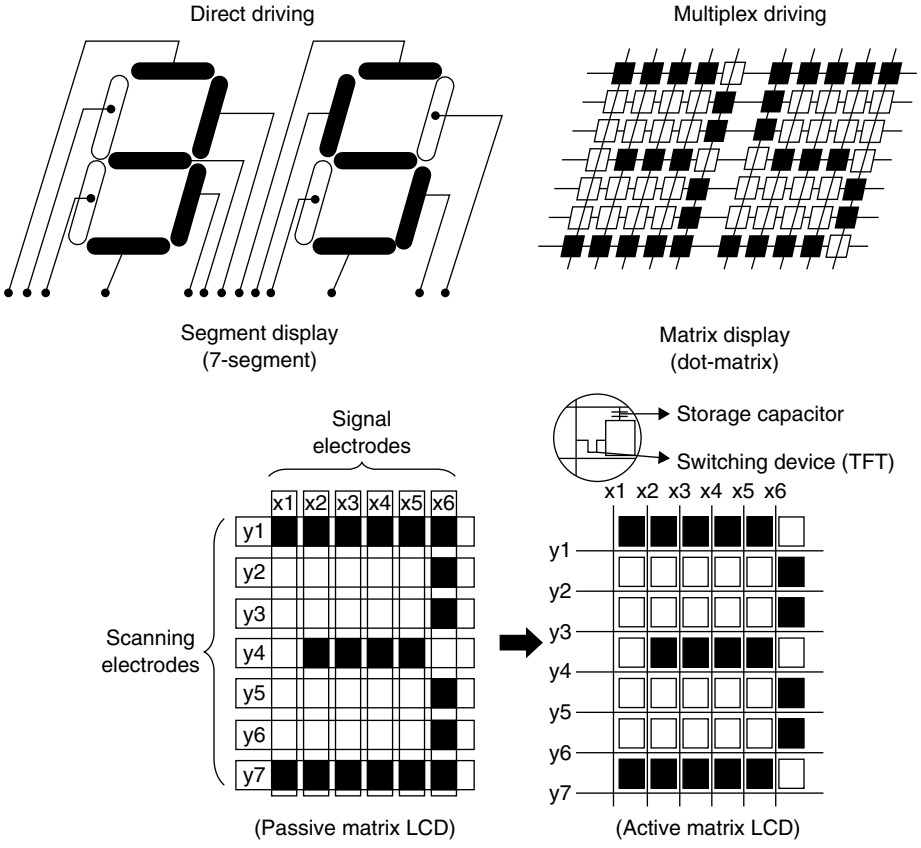


Fig.13.8 LCD control circuits

Advantages

- LCD has excellent resolution and hence the picture has good sharpness.
- It shows zero distortion of the images.
- The display has high-peak intensity that is capable of producing very bright images. This makes them best suitable for open air display.
- The screens are perfectly flat.
- They are thin and consume less electricity and produce less heat.

Disadvantages

- The pixel has a predefined resolution format at the time of manufacture that cannot be changed. This results in image degradation, particularly for fine text and graphics.
- Since LCD has an analog input there is a possibility of some interference from digital noise in the image. This digital noise cannot be eliminated completely.
- It has limited viewing angle that affects the brightness, contrast, gamma, and colour mixtures of the display.
- LCDs have trouble in producing black and very dark greys. This leads to lower contrast than CRTs and the colour saturation for low-intensity colours is also reduced. Hence they are not suitable for displaying information in dimly lit and dark environments.
- Images that are displayed in LCD are pleasing but not accurate because of their poor black-level, grey-scale, and gamma.
- LCDs that are matrix driven can have many weak pixels, which are permanently on or off. Some pixels may be improperly connected to adjoining pixels, rows, or columns. It may also show improper illumination from the backlight resulting in uneven intensity and shading over the screen.
- They are costlier when compared to CRTs.
- Speed of the displaying motion graphics is relatively slow.
- It has a narrow temperature operating range.

LIQUID CRYSTALS ON SILICON (LCOS)

In LCOS devices the crystals are coated over the surface of a silicon chip instead of the crystals and electrodes sandwiched between polarized glass plates. The electronic circuits are etched into the chip, which is coated with a reflective surface. Polarizers are in the light path before and after the light bounces off the chip. The construction of LCOS is shown in Figure 13.9.

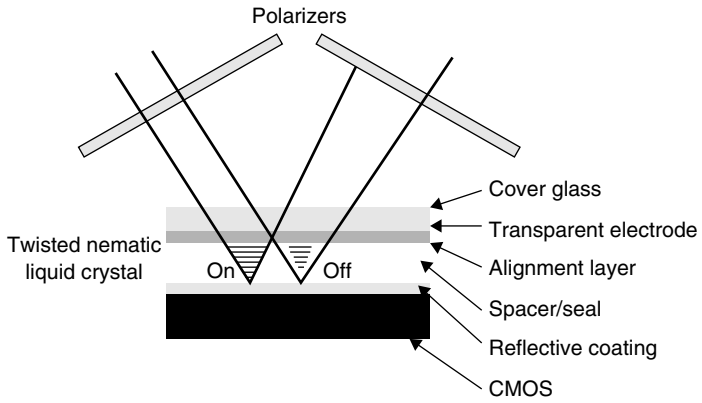


Fig.13.9 LCOS construction

Advantages

- LCOS are easier to manufacture than LCD displays.
- They have higher resolution since several million pixels can be etched onto one chip.
- The size of LCOS is much smaller. LCOS micro-displays are small and must be magnified via either a virtual imaging system or a projection imaging system.

Working of LCOS

The picture formation on an LCOS display involves various steps. The components of LCOS include a high-intensity lamp, a series of mirrors and micro-devices. The white light produced by the light source enters the condenser lens and is focused onto a filter. The visible light passes through the filter and this beam of light later passes onto the polarizing beam splitter. The polarizing beam splitter (PBS) divides the beam into three primary colors—red, green and blue. The beam then passes through a series of dichroic mirrors that permit only a desired wavelength of light and limits the other wavelengths. Each mirror permits the passage of only one of the three primary colors, while the other two primary colors are not allowed by it. Thus, these colored beams finally fall on the LCOS micro devices. There are individual micro-devices for each color, one for red, blue and green. The LCOS micro device is an important component of the display device. Unlike in the LCD where liquid crystals are sandwiched between two polarized panels, in LCOS the micro-device consists of a liquid crystal layer that is sandwiched between a transparent thin film transistor and a silicon semiconductor. The semiconductor chip is coated with a reflectant. The light from the source falls on the micro-device and the intensity of light falling on the reflecting surface of the micro-device is controlled by the liquid crystals that act as valves. The intensity of the light also depends on the applied voltage since the liquid crystals allow more light to pass through them at higher voltages.

Although the micro-device is explained as a single component, there are various layers in it. The instructions in the form of electricity, reach the printed circuit board.

Signals are sent to the silicon chip that acts as a sensor. Each pixel uses one transistor and is driven by the television pixel drivers. The image formed thus falls on the reflecting surface. Liquid crystal controls the intensity of light falling on it from the reflecting surface. The next layer comprises of a transparent electrode that connects the silicon and the liquid crystal. The entire set up is covered with glass cover for protection. When light falls on the reflective layer on these micro devices, they are reflected and finally pass through a prism where the red, blue, green light components combine. A complete image is formed and the prism directs it to the projector lens. The projector lens magnifies the image and displays it on the screen.

LIGHT EMITTING DIODES (LED)

The light emitting diodes are semiconductor materials. One must know the working of a diode to get a clear understanding of LED. A diode is a semiconductor material typically neutral. The purest forms of semiconductors are known as intrinsic semiconductors. The efficiency of these semiconductors can be improved by adding dopants (impurities) and the process is termed as doping. When a semiconducting material such as Ge doped is with a pentavalent metal such as gallium it forms a N-type material. The Ga has electrons in the outermost orbit to which four Ge electrons gets bonded. One gallium electron is free and the accumulation of these electrons of the dopants each time makes it a more negative and hence N-type material is formed. Similarly, when a trivalent material is added to Ge, results in positive or P-type semiconductor material. The Ga has three free electrons in the outermost orbit which gets bonded with the four Ge electrons but one Ge atom is unbounded. There exists a vacant space or hole. A hole is a negative deficit and therefore, considered as positive charge. Accumulation of holes results in a P-type semiconductor material. Therefore, N-type material has extra electrons and P-type has missing electrons or extra holes. A diode is formed by fusing an N-type and P-type semiconductor material thus forming a PN junction diode.

When these diodes are forward biased, i.e., the N-type is connected to the negative terminal of the battery and P-type to the positive terminal, then the diode works in positive biased mode and thereby conducts. This makes the hole from p type material and electrons from n type material to reach the junction forming the depletion zone. Free electrons moving across the diode are injected into the holes which creates a drop from conduction band to lower orbital thereby enabling the electrons to release energy in form of photons. The emitted light radiation is incoherent. The energy gap of these diodes determines the efficiency of the emitted light radiations, higher the energy gap, higher the spectral efficiency of the emitted photons. This method is called electroluminescence. These photons are to be allowed to emit without reabsorbing.

The recombination of electrons and holes can be classified into the following types:

- **Direct recombination:** In direct band gap materials, the lower energy of conduction band lies just above the maximum energy of the valence band which moves them to recombine with holes. The transition between conduction and valence band involves photon emission and is known as direct recombination. E.g. GaAs.

- **Indirect recombination:** In the indirect band gap materials, the minimum energy of the conduction band and valence band is slightly more which restricts the direct recombination. Thus to overcome this, many dopants are added that makes the donor states shallow. The donor states capture the free electrons and provide the necessary shift for recombination. This type of recombination is called indirect recombination. The colour and the wavelength of the emitted light depend on the band gap of the material. The equation that relates wavelength and band gap is given as:

$$\lambda = hc/E_g$$

Where h is Planck's constant, c is speed of light and Eg is energy band gap. The table shows the list of semiconductor materials and the corresponding colours.

LEDs can be made using two different doped semiconductor of the same material is called homojunction. When LEDs are doped using different materials they are called heterojunction and are much brighter than that obtained from homojunction.

The LED is manufactured such that all photons generated are emitted and not reabsorbed. The LEDs are capsulated inside a dome so that most of the light gets emitted efficiently. There are various shapes of domes available. The most common is round shaped.

Advantages of LEDs

- They produce more light than incandescent bulbs, thus useful in energy saving devices.
- They have the ability to emit light in a particular colour unlike other traditional light sources that use a colour filter for the same. This makes it more efficient and lowers the cost.
- When compared to incandescent and fluorescent sources that require reflectors to collect and direct light in an orderly manner, LEDs are designed to focus its light without any additional components.
- The brightness of LED can be lowered by altering the current flow. The colour of the LED remains the same unlike the incandescent source that turns yellow at low voltages.
- The LED responds quickly to ON and OFF conditions when compared to the traditional light sources that require longer restarting time.
- LEDs being solid state components withstands the fluctuations in supply voltage.
- They possess longer usage period for about 50,000 hrs than a fluorescent tube light that is stated to have about 30,000 hrs and incandescent bulbs that have 1000–2000 hrs.
- They do not contain mercury unlike the traditional lighting.
- As they are small they can be accommodated in printed circuit boards and reduces the size of the display devices to greater extent.

Disadvantages

- They are expensive than the conventional light sources.
- Heat dissipation is a major issue concerning LEDs. Therefore, ambient temperature must be maintained for good functioning of LED. If the temperature goes high, it may get heated and may cause failure of the device.

- As the LEDs efficacy depends on the current supply, an appropriate amount of current must be supplied which can be controlled by electronic components like resistors and current regulated power supply thereby making the circuit complex.
- The light from LED cannot be focused on a point thus it cannot be used for applications that require highly collimated beam.

LED Display

A LED display consists of many pixel modules. Each pixel module is made up of 4+ LEDs of RGB. Since there are several wires that run to each LED module, the display circuit board appears clumsy and complex. As there are numerous LEDs used in a display unit the power supply is also high.

7-SEGMENT DISPLAY

A 7-segment display (SSD) is a simple electronic display device that can be used to display decimal numbers and alphanumeric. It is basically a figure of numeral eight formed by grouping of LEDs. Some include a decimal point (DP). The 7-segment display comprises of seven LEDs with individual LED forming a segment. Each segment is labeled starting from alphabet 'a' through 'g'. The segments can be operated either individually or can be interconnected to display alphanumeric. The seven segments are placed such that they form a rectangle of two vertical segments on each side with one horizontal segment on the top, middle, and bottom as depicted in Figure 13.10. The seventh segment bisects the rectangle horizontally. Other segment displays like 14-segment displays and 16-segment displays can be used for full alphanumeric. The figure is a typical component and pin layout for a 7-segment display.

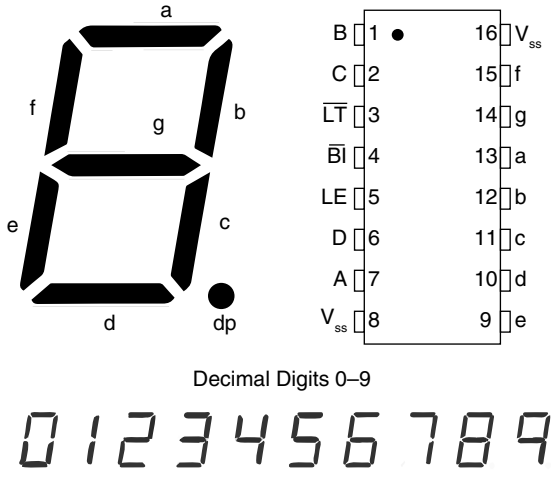


Fig.13.10 7-segment display—pen layout

Construction and Working

The light emitting diodes can be placed in a particular manner to obtain the 7-segment display as shown in Figure 13.11.

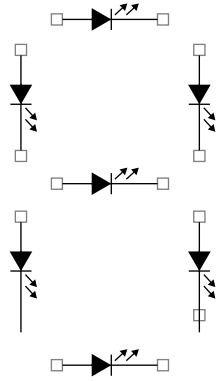


Fig.13.11 7-segment display using diodes

The display can be classified into two types based on the way the LEDs are connected with each other. They are:

1. Common cathode, where the cathode of all LEDs are connected.
2. Common anode, where the anode of all LEDs are connected.

The diode connections for common cathode and common anode are shown in Figure 13.12.

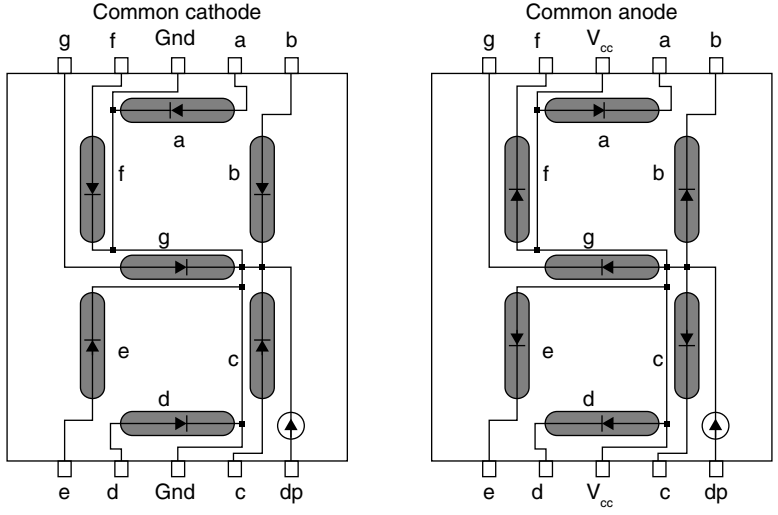


Fig.13.12 Common cathode and common anode LED displays

Binary to signal conversion can be carried out in an LED only when the LED is driven with a display driver. The commonly used display driver chip is MC14511-b is shown in Figure 13.13.

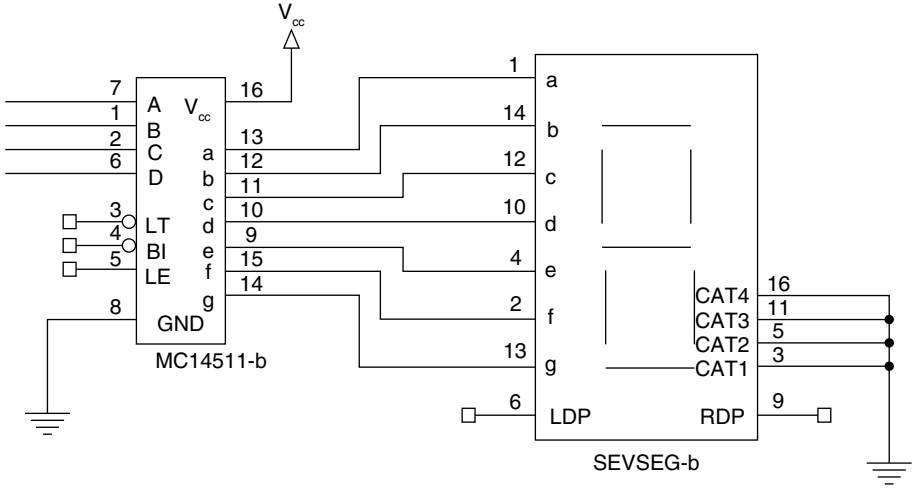


Fig.13.13 Display driven chip MC14511-b

Truth Table for binary to signal conversion using LED

Inputs							Outputs							
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	8
X	0	1	X	X	X	X	0	0	0	0	0	0	0	Blank
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	1	1	1	1	1	0	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	Blank
0	1	1	1	0	1	1	0	0	0	0	0	0	0	Blank
0	1	1	1	1	0	0	0	0	0	0	0	0	0	Blank
0	1	1	1	1	0	1	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	0	0	0	0	0	0	0	0	Blank
0	1	1	1	1	1	1	0	0	0	0	0	0	0	Blank
1	1	1	X	X	X	X	*							*

X = Do not care

*Depends upon the BCD code previously applied when LE = 0

A, B, C, and D are the binary inputs; a, b, c, d, e, f, and g are the driver signals to the display elements. LT is the light test control, turns all segments on, active low. BL blanks all the segments when activated, active low. LE is the latch-enabled control. The truth table shown below is used to check whether the digital signal sent to the display lights up the correct segment. The truth table for the binary to signal conversion is as shown on the previous page.

The internal circuitry of the conversion is shown in Figure 13.14.

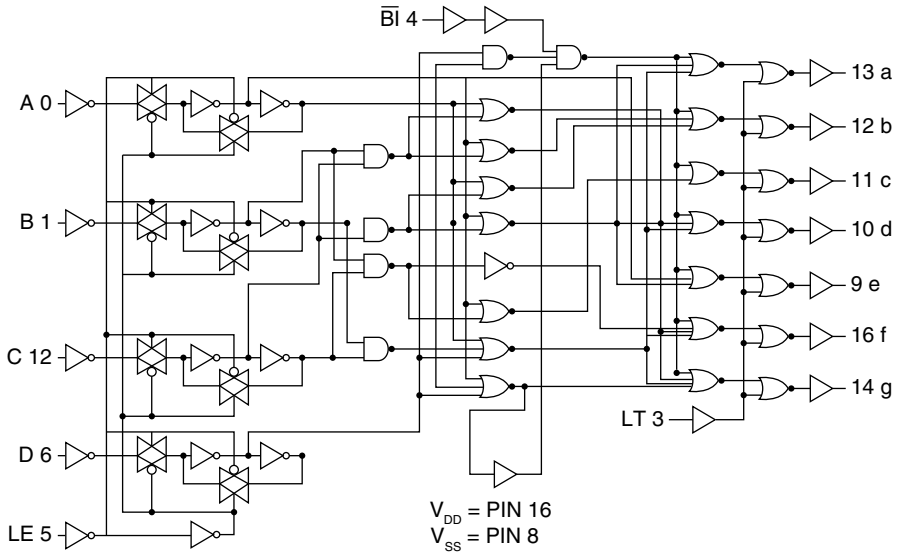


Fig.13.14 Circuit for 7-segment LED to BCD conversion

INTRODUCTION TO ORGANIC LIGHT EMITTING DIODES (OLED)

Organic light emitting diodes (OLED) have attracted considerable attention over the past decades owing to their promising applications in the field of flat panel displays. Their thin screens and high-colour rendition have made a better alternative to cathode ray tubes (CRT) and liquid crystal displays (LCDs). OLEDs was first reported in 1960 from single crystals of anthracene and later in 1987 Tang and Van from Kodak achieved a breakthrough in discovery of OLEDs. They discovered efficient and low voltage OLEDs from p-n hetero structure devices using thin films of organic materials.

OLEDs are a class of light emitting diodes that work on the principle of electroluminescence. Electroluminescence is the process of emission of light from organic materials when an electric field is applied to them. Electroluminescent materials are of two types, small molecules organic light emitting diodes (SMOLED) and polymer light emitting diodes (PLED). The electroluminescent performance of both SMOLED and

PLED are similar but the deposition of the organic material is by different mechanisms. In small molecules, organic light emitting diodes the deposition is by evaporation under vacuum referred as dry process and in case of PLED they are processed by solution and thereby referred as wet process. The construction of SMOLED is much complex and sophisticated than PLED.

Components of OLED

- **Substrate:** It consists of a substrate that is generally made of transparent plastic or glass and forms the base of the OLED.
- **Anode and cathode:** OLED consists of an anode and a cathode. The anode is transparent and is made up of indium tin oxide (ITO) as it has good transparency at the visible range, while the cathode is reflective and is made up of metals with low-work function like Ca, Mg, Ag, Al, etc. Light is produced by recombination of holes and electrons that are injected in the electrodes. The charge carriers in an anode are holes. Holes are positive charge and can be explained as an absence of electron. Cathode can be made up of materials that may or may not be transparent. The cathode releases electrons when current is passed to it.
- **Organic layer:** The organic layer may be small molecules or polymers. The thickness of the organic material is about 100–150 nm.
- **Conducting and emissive layer:** The conducting layer is made up of organic plastic that help in the movement of holes from the anode. Polyaniline is an example for conducting polymer used in OLED. The emissive layer is made up of organic plastic molecules that transport electrons from the cathode and is different from the conducting layer. The electrons and holes recombine and light is given in this layer. Polyfluorene is a commonly used polymer for emissive layer. The tedious part of manufacturing OLEDs is applying the organic layers to the substrate.

Construction of OLED

OLED consists of a thin film of organic material sandwiched between two electrodes. The organic material behaves like an insulator. The materials used in OLED are amorphous or semi-crystalline films. SMOLED uses derivatives of triarylamines for p-type material and derivatives of metal chelates like tris (8-hydroxyquinolato) aluminium (III) Alq3, triazoles, or oxadiazoles for N-type material. Phosphorescent materials have been used in PLED. The efficiency of the emitting layer can be improved by doping it with various organic dyes. Doping rate is about 1–2 wt% and this solution has been widely used to enhance the colour and device lifetime. The main requirements for OLED material are high luminescence, good charge mobility, good thermal and oxidative stability, and excellent colour purity. One of the characteristics of an OLED is the pixel that is an emissive device that can be switched off and be completely black when compared to a liquid crystal where the pixel is a transmissive device which does not allow complete occultation of backlight.

Depositing the organic material onto the substrate to obtain red, green, and blue pixels is a major challenge as it requires accurate poisoning and uniformity in deposition. There are three ways of depositing the organic layer to the substrate:

1. Organic vapour phase deposition (OVPD).
2. Inkjet printing.
3. Vacuum deposition or vacuum thermal evaporation (VTE).

Small molecules are deposited by evaporation through shadow mask and polymers are done by inkjet printing. Conventional techniques like lithography cannot be applied as the materials do not withstand the process and the layers obtained are very thin. The common techniques used for deposition of the organic material onto the substrate are discussed below.

Vacuum Deposition or Vacuum Thermal Evaporation (VTE): The process of depositing the organic material onto the substrate by vacuum deposition technique can be explained by the following steps:

- The small molecules are placed in crucibles and heated up to 100–500°C
- The shadow masks of thickness 20–100 μm is placed above the crucible. The masks have holes to hold one-third pixels.
- A stack of single colour is deposited and shifted by one pixel and the next colour is deposited.
- The process is repeated until the substrate is completely deposited.

Care is taken while handling large shadow masks as their positioning is to be done with a precision of ±5 μm. As each colour has a different lifetime, the display colour becomes unbalanced. This leads to a further limitation.

Inkjet Technique: Steps involved in depositing the organic material on the substrate by inkjet technique is discussed below.

- The inkjet printing technique involves dispensing polymer materials that are soluble in solvent.
- The solution is dispensed by inkjet nozzles on the substrate.
- The drops measuring a few picolitres are injected accurately by the inkjet head.
- Polyamide banks are built around the pixel area forming a well.
- These banks are water repellent and the pixel is hydrophilic to prevent sticking on the banks.
- The pixels are filled properly and once the deposition is complete the droplets are dried, the solvent is evaporated and the film is formed.

Working of OLEDs

The voltage is applied to the OLED from the battery. When a voltage is applied between the electrodes, charges are produced. Holes from anode and electrons from cathode are injected into the organic material. During the recombination process electrons recombine with the holes at the boundary of the emissive layer. Charges

move inside the material and form excitons. The recombination zone depends on the charge mobility of the organic material and the strength of the electric field applied. Hence luminance is proportional to current density.

After diffusion, exciton recombines and photon is emitted. The colour of the photon depends on the energy difference between the highest occupied molecular orbit and lowest unoccupied molecular orbit. In simple words, it depends on the organic materials used. The wavelength can be controlled by the extent of conjugation in the molecule or polymer. The OLEDs work in two different modes of operation:

1. **Active matrix displays (AMOLED):** AMOLEDs are packed with layers of cathodes, organic molecules, and anodes. The anode has a thin film transistor (TFT) arrangement that forms a matrix. The transistor circuit controls each pixel which turns it ON to get the image. AMOLEDs consume less power than PMOLEDs as transistor circuitry requires less power than external circuitry and hence they are efficient for large displays. A large number of transistors are involved in controlling a large number of pixels in high resolution or large displays. The restore time is faster for AMOLEDs which makes them suitable for video displays. These types of displays are used in computer screens, electronic billboards, and large TV screens.
2. **Passive matrix displays (PMOLED):** PMOLEDs consist of strips of anode and cathode. The anode and cathode strips are arranged perpendicular to each other. The intersections of the cathode and anode make up the pixels where light is emitted. The pixels are controlled by an external circuit that decide the ON/OFF state of the required pixel. PMOLEDs are easy to make, but they consume more power than other types of OLED but definitely less than an LCD display. The power consumption is due to the power needed for the external circuitry. PMOLEDs are most efficient for texts and icons and are best suited for small screens like cell phone panels and MP3 players.

Types of OLEDs

- **Transparent OLEDs:** Transparent OLEDs have only transparent materials for the making of all their components (substrate, cathode, and anode). A transparent OLED allows the emitted light to pass in both the directions when it is turned ON. They work with both active-matrix and passive-matrix modes of displays.
- **Top-emitting OLED:** These types of OLEDs have their substrate made up of an opaque or reflective material. They emit light only in one direction and work best in the active-matrix display mode.
- **Foldable OLED:** Foldable OLEDs have substrates made of very flexible metallic foils or plastics. They are lightweight and long-lasting. Since the substrate is a flexible material they reduce breakage and hence repair.
- **White OLEDs:** White OLEDs emit white light that is brighter, more uniform, and more energy efficient. These are more efficient than fluorescent lights. They show excellent true-colour qualities. The white OLED have started to replace fluorescent lights that are currently used in homes and buildings as they consume less energy and cost lesser.

Lifetime and Device Stability

Device stability is an important issue for any emissive technology such as OLED and particularly when it comes to the ageing of the three primary colours. The lifetime can be simply defined as the mean time to half brightness. The display applications generally have a lifetime over 20,000 hours. Degradation occurs both under operating conditions and while it is used as storage device. The process of degradation is by dark spot degradation, catastrophic failure, and intrinsic degradation.

Advantages

The key advantages of OLED for flat panel display applications are their

- Self-emitting property
- High luminous efficiency
- Colour rendition capability
- Wide viewing angle
- Light weight, transparency, and flexibility
- High contrast
- Low-power consumption
- Potentially large area colour display

Disadvantages

- OLED have colour imbalance due to lifetime of organic materials. The red and green OLED films have longer lifetimes (46,000–230,000 hours) while the blue films have much shorter lifetimes (up to around 14,000 hours). This leads to the non-uniformity of the films as each has a different lifetime thereby creating a colour imbalance and poor image quality.
- Processes involved in the manufacturing of OLED are expensive and time consuming.
- They possess poor water resistivity and hence can be easily spoiled when water spills on its surface.

Applications: OLEDs are used in small-screen devices such as cell phones, PDAs, and digital cameras. Rapid progress in OLED research and development has led to flexible display screen known as flexible organic light emitting diodes (FOLED). Instead of glass surfaces as display screens, the FOLEDs are made on flexible substrates (transparent plastic to opaque metal foils). OLEDs have faster response time than LCDs and hence video images could be much more realistic and constantly updated.

PLASMA DISPLAYS

The first plasma displays that were available were technically televisions, because they did not have TV tuners. The television tuner is the device that converts a television signal from a cable wire or any other means into a video image. Since the plasma

displays lacked the tuners, they were used as simple monitors that display a standard video signal. Recent developments have made inbuilt digital television tuners. Plasma displays are flat panel that use gas discharge as their working principle. Owing to their slim display size about 6 inches (15 cm) they have substituted the CRTs in the television monitors. Plasma is a gas that is ionized and electrically conductive, made up of free flowing ions and electrons. It was noticed that some gases glowed when an electrical current was supplied to them. The voltage at which the gas inside the tube becomes plasma is known as its "breakdown" voltage (typically 100–200V). This voltage varies with the composition and pressure of the gas. Plasma displays commonly use noble gases like neon, argon, and xenon. The actual gas in the plasma display is often a mixture of these noble gases. Sometimes to improve the colour rendition, "dopants" or impurities are added to the gases.

Working of Gas Plasma Displays

In general, any gas is mainly made up of uncharged particles at normal conditions. This is because the number of protons (positively charged particles in the atom's nucleus) and electrons (negatively charged electrons) are equal. The negatively charged electrons perfectly balance the positively charged protons, thereby the atom has a net charge of zero. When an external electrical voltage is supplied to the gas, it ionizes the gas and introduces free electrons. Ions are electrically charged atoms. The free electrons collide with the atoms and tap the other electrons. In the process, electrons are lost which creates an imbalance in the charge of the atom. Thus, loss of electrons results in increasing the net positive charge making it a positive ion. The electrical current keeps running through the plasma making the negatively charged particles move towards the positively charged area of the plasma, and positively charged particles are rushing to the negatively charged area causing collisions between them. These collisions excite the gas atoms in the plasma and cause them to lose the energy gained from ionization in the form of visible light. The intensity and colour of the light depends on the gas used, its pressure, and the amount of electrical current flowing through it.

The light is emitted only when the gas is partially evacuated and not for high or low pressures of the gas. This pressure gives information on the density of the gas. If the gas is at high pressure then too many molecules are present per unit volume which does not allow the gas to glow. This is same for the gas maintained at low pressure. Though the working principle is simple compared to the other display technologies, implementation into high-resolution display panel is complex. Plasma displays are currently competing with other technologies for the lucrative military and industrial display market. The flat profile of the plasma panel fits into tight spaces such as military vehicles.

Types of Gas Plasma Displays

Plasma displays are currently made in two forms, DC plasma and AC plasma displays, based on the type of electrical voltage applied to them.

DC Plasma Display: The most basic type of gas plasma display is the DC plasma. It consists of two electrodes anode and cathode. The anode and cathode are in the form of strips and are placed perpendicular to each other. The anode is made from a transparent oxide coating on the top surface of the display, or from a wire mesh. The DC voltage applied between the anode and a cathode, ionizes the gas between them. The anodes are placed as segments to form a segmented numeric or alphanumeric display. The gas is in direct electrical contact with the electrodes of the cell. They do not possess any external pixel memory. Most DC plasma displays are dot matrix. The display can be made to work by making one cathode conduct at a time. By controlling the anode voltages, dots are formed at the desired intersections. The display size is limited as it is directly proportional to the time duration of each dot in the ON state. Brightness decreases as display complexity increases. This type of display also requires one driver for each row and each column. Unlike LCD displays, the drivers must be able to switch high voltages, which make them expensive. The main factors affecting speed in larger panels are the configuration of the matrix and the method used to apply the information to the matrix. In large plasma displays, the matrices are often subdivided into a number of smaller and more quickly managed sub-matrices. The more the entire matrix is subdivided, the smaller each sub-matrix becomes, and the overall display can be operated faster.

AC Plasma Display: The AC plasma panel operates on AC voltage. In the AC display panel, insulating layers are present between the electrodes. This is the major difference between the AC and DC display panels. The dielectric layer forms a capacitor. A continuous AC voltage below the threshold voltage is applied to all the electrodes which ionize the gas. This voltage keeps the display on. The pixel memory is the prime advantage of the AC plasma cell over the DC type. AC plasma display therefore does not require refresh, and a separate refresh memory is not required as with other multiplexed displays. The circuitry involved in AC plasma display is complex and must control high voltages, resulting in a relatively expensive display.

Advantages

- They have excellent picture quality.
- The image appears clearer and sharper compared to CRT and LCDs.
- They have wider viewing angle and uniform brightness.
- High-colour rendition levels with high contrast ratio.
- Plasma displays have low-power consumption.

DIGITAL LIGHT PROCESSING

Digital light processing (DLP) has revolutionized the display technology. It is based on the digital micro mirror device (DMD). DLP creates links to the final display information. DLP has three key advantages over existing projection technologies. DLP is noise-free and has accurate image quality with digital grey scale and colour

production. The micro mirrors are placed very close to each other, thus making the resolution better and the size of the display panel small. A digital micro mirror device is shown in Figure 13.15.

The central, reflective portion of the device consists of 508,800 tiny, tiltable mirrors. A glass window seals and protects the mirrors.

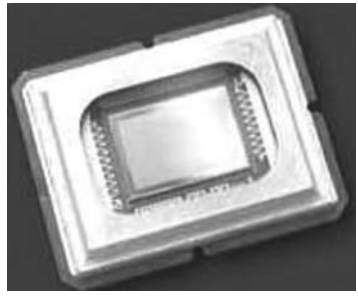


Fig.13.15 An 848 × 600 digital micro mirror device

Working of DLP

DMD forms the basis of digital light processing. DLP can be made of one, two, and three-chips. A DLP-based projector system has the following components like memory and signal processing unit, light source, a colour filter, cooling system, and illumination and projection optics. A DMD is a semiconductor light switch that consists of square-shaped tiny mirrors. Each mirror is capable of switching a pixel of light. A glass seals and protects the mirrors. DLP system chooses a digital or an analog signal depending on its application. The signal goes through DLP video processing and becomes progressive red, green, and blue (RGB) data. The RGB data is then formatted into entire binary bit of data. Each pixel is connected to a single mirror. If the signal is 640 × 480 pixels, the central 640 × 480 mirrors on the device will be active and all other mirrors outside this area is OFF. The mirrors are capable of switching ON and OFF more than 1000 times a second. This rapid speed allows digital grey scale and colour reproduction. DLP is considered as a simple optical system where light from the projector lamp passes through condensing optics and a colour filter system and reaches the DMD. When the DMD is in the ON position, the mirrors reflect the light falling on it to the display screen. The working of digital light processing is shown in Figure 13.16.

Advantages of DLP

- It is capable of giving a final link and completes a digital video system.
- The colour levels are high and the exact picture can be recreated.
- Since DMD is reflective device, the light efficiency is greater and it does not require polarized light.

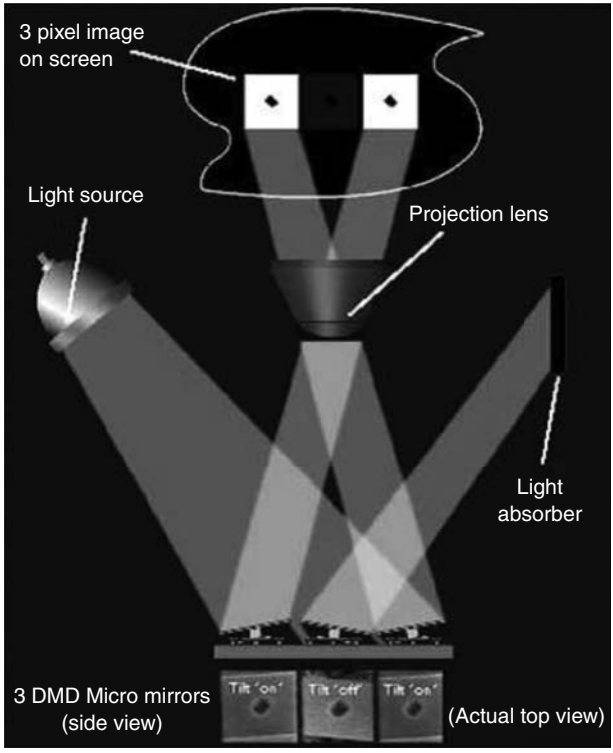


Fig.13.16 Working of digital light processing

- The square mirrors of DMD are 16×16 μm^2 with a gap of 1 μm between each mirror. They reflect 90% of the light falling on them. Since they are placed very close, their resolution is clearer and higher.
- They have greater reliability against environmental and operational tests.
- DMD are thermal, mechanical, temperature, and moisture resistant.

The DLP has excellent optical system that is operated by digital signals. A video or graphic signal is used to create a digitally extraordinary picture quality. Because it is based on the reflective DMD, DLP is more efficient than LCD technologies.

E-BOOKS/E-INK

The inventions of display technology has grown into leaps and bounds over the years. Although there are different displays like LED, LCD, OLED, gas plasma still paper is one of the best inventions till date. Paper being the simplest and effective way of displaying information has rekindled the invention of e-paper. Scientists have developed a technology that could replace paper, called electronic ink or e-ink. Electronic paper is a general name given for display technologies that resembles real paper. The e-ink

offers papers in black and white. E-ink is the leader of e-paper displays. It works on the electrophoretic principle. They are made up of millions of tiny balls or dots 0.04 mm in diameter containing black and white particles in them. The black balls are negatively charged, while the white balls are positively charged and both keep floating inside the capsule. The movement of the particles can be influenced by electric field. The white or black particles can be chosen to move inside the capsule by adjusting the electric field. The black or white particles can be aligned in the top portion or bottom portion of the capsule with the help of this electric field thereby changing the colour of the pixel. The position of the particles can be retained in the absence of the electric field. Since the colour of the display is not changed, it requires low power for its operation. The working is depicted in Figure 13.17. Coloured e-papers can be obtained by placing colour filters on top of the black and white particles.

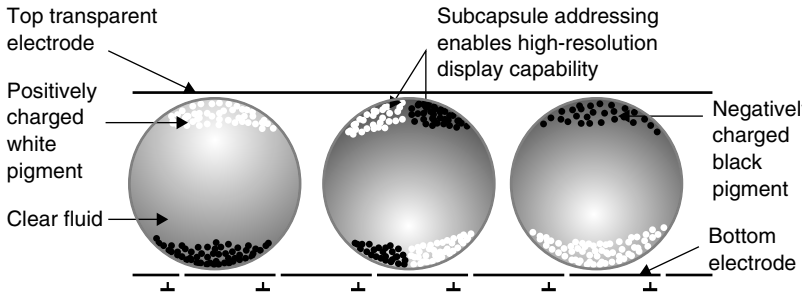


Fig.13.17 Formation of E-ink

Advantages

- E-paper uses low power as they do not require to be refreshed.
- They possess high contrast.
- It requires no backlight and can be viewed under direct light.
- They are thin and flexible.
- Electronic ink can be printed on any surface.
- Its excellent readability makes it appear like printed text and does not cause strain on eyes.

MULTIPLE CHOICE QUESTIONS

1. The grid in the vacuum tube is maintained at _____ potential.
   (a) Positive (b) Negative
   (c) Zero (d) 220v
2. Electrons from the cathode of the CRT are emitted due to _____.
   (a) Secondary electron emission (b) field emission

(c) thermionic emission (d) radiation

3. The color display monitors have colored dots coated with _____ its rear side.
   (a) Sodium (b) Fluorescent
   (c) Phosphor (d) Zinc
4. In color CRT display, the various colors are obtained by mixing _____ colors at different intensities.
   (a) 2 (b) 3
   (c) 16 (d) million
5. The primary colors present in color raster display are
   (a) red ,gray, blue (b) red, green, beige
   (c) red, gray , beige (d) red, green, blue
6. A color CRT has _____ electron guns.
   (a) 1 (b) 2
   (c) 3 (d) 4
7. The quality of the displayed image can be improved by
   (a) Increasing the number of pixels (b) Decreasing the number of pixels
   (c) Applying high current (d) None of the above
8. The memory device used to store the color or data information is called as
   (a) RAM (b) ROM
   (c) Frame buffer (d) None
9. In nematic liquid crystals the twisting of crystals to certain angle depends on
   (a) Orientation of the crystal (b) Applied voltage
   (c) Applied pressure (d) Applied current
10. A polarizing filter allows only light rays that are _____ to the lines on the filter.
    (a) Parallel (b) Perpendicular
    (c) Random (d) dispersed
11. The electrodes in a LCD are
    (a) Translucent (b) Transparent
    (c) opaque (d) non conducting
12. In liquid crystal on silicon (LCOS) display, the crystals are
    (a) sandwiched between polarized glass plate
    (b) electrodes are sandwiched between glass plate
    (c) coated on silicon chip
    (d) Coated with reflective surface.
13. Dichroic mirror in LCOS helps to
    (a) Reflect few wavelengths and transmit the rest of the light
    (b) Splits the rays falling on it
    (c) Polarizes the rays falling on it

(d) Only reflects and does not transmit

14. Light emitting diodes work on the principle of
    (a) Electroluminescence (b) Photoluminescence
    (c) Thermo luminescence (d) None of the above
15. LED's fabricated from two different types of doped semiconductors of the same material are called as
    (a) Heterojunction (b) Homojunction
    (c) Polyjunction (d) Monojunction
16. The letter "p" in a p type semiconductor refers to
    (a) Passage (b) Positive
    (c) plasma (d) photon
17. The color emitted by the LED depends on the
    (a) Color of the dome (b) Semiconductor material
    (c) Voltage applied (d) Number of photos emitted
18. The technique of depositing the organic material through evaporation under vacuum is used in
    (a) SMOLED (b) LED
    (c) LCD (d) PLED
19. PLED stands for
    (a) Photon light emitting diodes (b) Positive light emitting diodes
    (c) Plasma light emitting diodes (d) Polymer light emitting diodes
20. The light is produced in LED due to
    (a) Recombination of electrons and holes (b) Presence of charge carriers
    (c) The reflective coating on the electrodes (d) None of the above
21. Which of the following is a commonly used material for emissive layer in OLED
    (a) Polyaniline (b) Phosphor
    (c) Polyfluorene (d) polyamide
22. The light emitted in plasma gas display is due to
    (a) Ionization of gas (b) High Pressure of the gas
    (c) High voltage applied (d) All of the above
23. The plasma displays can be fabricated as
    (a) AC plasma display (b) DC plasma
    (c) Both (d) None
24. The display in digital light processing is due to
    (a) Repeated reflections (b) Repeated refractions of light rays
    (c) Digital mirror (d) Color filter
25. The principle behind the working of e-ink is
    (a) Electrophoretic (b) Piezoelectric

(c) Photoelectric (d) thermo luminescence

26. The movement of tiny particles in E-ink is influenced by
    (a) magnetic field (b) electric field
    (c) both electric and magnetic fields (d) none of the above

Answers

1. (a)	2. (a)	3. (c)	4. (c)	5. (d)	6. (b)	7. (a)	8. (c)	9. (b)
10. (a)	11. (a)	12. (c)	13. (a)	14. (a)	15. (b)	16. (b)	17. (b)	18. (a)
19. (d)	20. (a)	21. (c)	22. (d)	23. (c)	24. (c)	25. (a)	26. (b)	

DESCRIPTIVE QUESTIONS

1. Explain the construction and working of liquid crystal displays.
2. Briefly explain the types of LCD devices.
3. Construction of organic light emitting diodes (OLED).
4. Draw and discuss the working of cathode ray tubes (CRT).
5. Explain the working of a 7-segment display with a neat circuit diagram.
6. Mention the advantages and disadvantages of LED.
7. Discuss the modes and types of OLED.
8. Mention the advantages and disadvantages of OLED.
9. Explain the working of gas plasma displays. Write a brief note on its types.
10. Discuss the working of digital light processing.

CONVERTORS

Digital fields processes signals, like temperature, pressure, and light. The processing of these signals requires accurate signal conditioning and signal conversion. These processing systems form an integral part of instrumentation and high-performance integrated circuit (IC) technology. High precision processing requires a low noise ratio. Thus, to create these devices is a challenge in today's applications the analog conversion is needed. The most common analog-to-digital conversions (ADCs) and digital-to-analog conversions (DACs) are described in detail in this chapter.

ANALOG-TO-DIGITAL CONVERSION

Analog-to-digital convertor is an electronic integrated circuit which transforms a signal from analog (continuous) to digital (discrete) form. They are often abbreviated as ADCs. Digital signals operate at two states and are always referred in terms of binary states, 0 and 1. Analog signals are varying signals that can be measured directly. Physical quantities like temperature, pressure, velocity, or electrical quantities like voltage, current respond to the analog system and produces proportional signals that can be measured. These signals are displayed in the analog meters as deflection. The deflection directly relates to the quantity measured. The deflection is a percentage of the full scale deflection. This chapter first explains an ADC and describes typical ADC building blocks followed by DACs. An analog-to-digital consists of the following stages as represented in Figure 14.1.

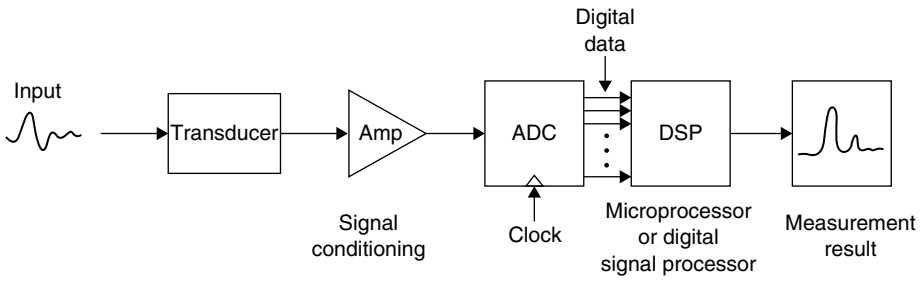
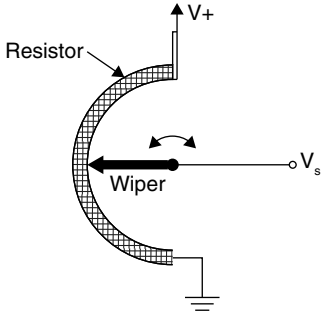


Fig.14.1 ADC building blocks

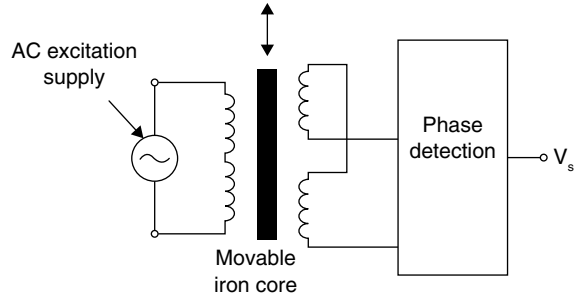
A bio signal must be converted to an electrical signal by passing it to the transducer. Transducers are devices that are capable of converting one form of energy into another. Transducers are of many types as shown in Table 14.1.

Table 14.1 Types of transducers and variables converted

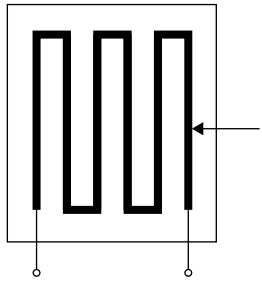
Transducer type	Variable converted
Potentiometer	Position
Strain gauge	Pressure
Piezoelectric	Force
Thermistor	Temperature
Thermocouple	Temperature
Photoconductive	Light
Photo transistor	Light
Field effect transistor	Current
Microphone	Sound



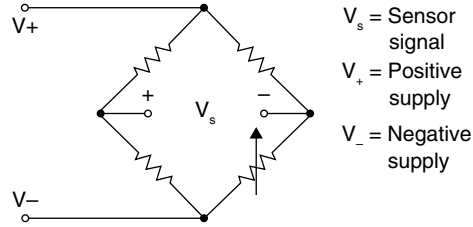
(a) Potentiometer



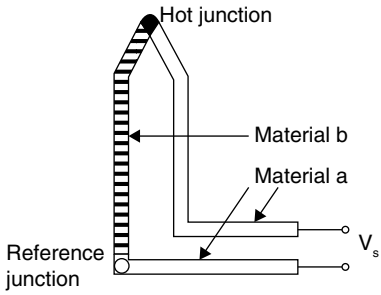
(b) Linear variable displacement transformer



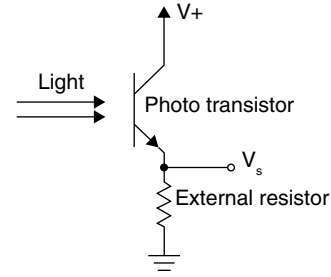
(c) strain gauge pad



(d) strain gauge circuit



(e) Thermocouple



(f) Photo transistor

The output from a transducer must be further filtered as it is unrefined and cannot be used directly. The output from the transducer is passed through the filtering circuit which is constructed using operational amplifiers to form a low pass filters. Filtering helps in enhancing signals. Finally, this output is fed to the ADCs and stored in the computer.

There are various types of ADCs, namely, single- and mixed-signal devices. Single-signal devices either use analog or digital signals. Mixed-signal device has both analog and digital functions and since it uses both the forms of signals it is considered to be complex. An ADC generally converts an input voltage or current to a digital word. The analog input to be converted is compared with reference voltage or current. The digital output is expressed as a fraction of the reference voltage or current depending on the input voltage or current. The input/output transfer function is given by the formula:

$$\text{Output} = 2n \times G \times A_{in}/V_{ref}$$

where,

n = Number of output bits also refers to resolution
G = Gain which is usually "1"
A_{IN} = Analog input voltage (or current)
V_{ref} (I_{ref}) = Reference voltage (or current)

Principle of ADC

The working of an ADC can be explained with the help of an example. Consider a 3-bit ADC; it has $2^3 = 8$ possible output codes. The difference between each output code is calculated as $V_{REF}/2^3$. On increasing the input voltage by 1 V, the output code will increase by 1 bit. Assuming the output response has no errors, the increase in input voltage by steps of 1 V shows an increase in the output code by 1-bit. The least significant bit (LSB) always corresponds to 1 V. Therefore, the resolution of the convertor is 1 V as it can resolve voltages as small as 1 V. Resolution is also expressed in bits. If the voltage is reduced by 0.8 V instead of 1 V then the LSB corresponds to 100 mV. This enables the convertor to measure even smaller voltage

range with greater accuracy. This method is adopted for achieving better precision and higher resolution convertors without spending much.

$$1\text{LSB} = \text{step voltage} = (\text{voltage range})/2^n$$

$$\text{Resolution (V)} = \text{FSV}/(2^n - 1)$$

Therefore, the resolution of ADC depends on the output bits.

ADC PROCESS

The analog-to-digital conversion process has four main steps:

1. **Sampling and holding (S/H):** The sample and hold component is the most essential part of the ADC. It enables to hold the analog input at constant during the process of conversion. Holding signal benefits the accuracy of the ADC. The sample pulses should have a minimum sampling rate, i.e., at least twice the highest data frequency of the analog signal as shown in Figure 14.2.

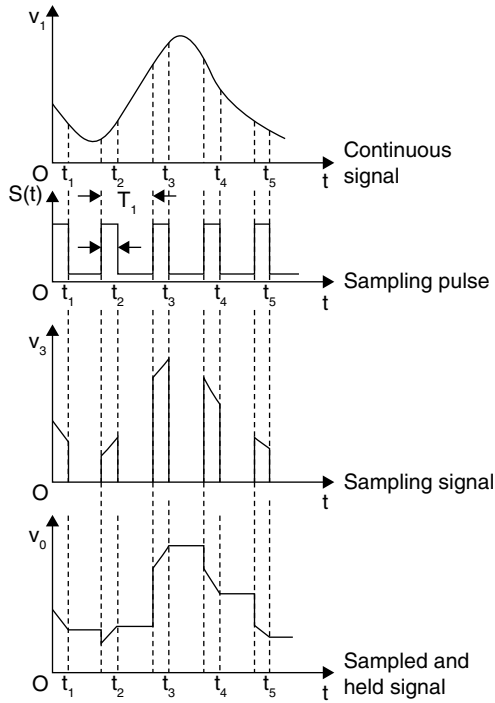


Fig.14.2 Comparison of sampled pulses and analog signal

2. **Quantizing and encoding (Q/E):** Quantization is the process of breaking down the continuous analog signal to discrete output states. Each discrete state is in binary form and represents a particular value. The widths of these steps are equal.

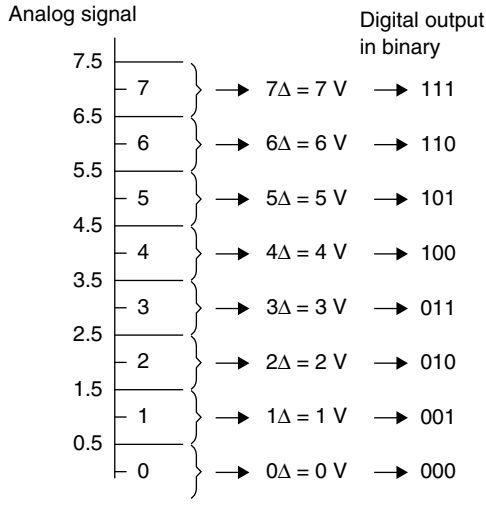
The quantization error is defined as the difference between digital output and analog input.

3. **Resolution:** The smallest change in analog signal that will result in a change in the digital output is called resolution. The resolution represents the quantization error inherent in the conversion of the signal to digital form

$$\Delta V = \frac{V_r}{2^N}$$

where,
ΔV = Resolution
V_r = Reference voltage range
2^N = Number of states where N is the number of bits in digital output

4. **Encoding:** This process assigns a unique digital code to each quantum and later allocates the digital code to the input signal.



$\Delta V = 1$ V

Maximum quantization error = ± 0.5 V

ACCURACY OF A/D CONVERSION

The accuracy of ADCs can be improved by two ways:

1. The resolution of the convertor must be increased. This improves the accuracy in measuring the amplitude of the analog signal.
2. The sampling rate must be increased which increases the maximum frequency that can be measured as shown in Figure 14.3.

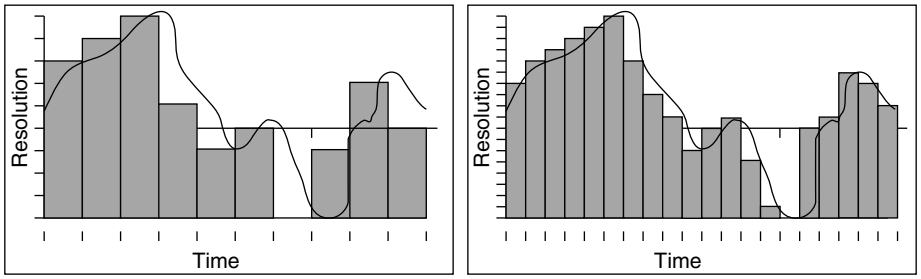


Fig.14.3 Sampling rate dependency

TYPES OF A/D CONVERTERS

- Digital ramp or staircase ramp
- Successive approximation ADC
- Flash ADC
- Delta-sigma ADC
- Dual slope ADC
- **Others:** Voltage-to-frequency, or single slope, charge balancing or redistribution, switched capacitor, tracking, etc.

DIGITAL RAMP ADC

Digital ramp ADC is the simplest form of analog-to-digital convertors. It consists of a binary counter that works as a register. The register is controlled by a clock input through an AND gate. The clock keeps incrementing the counter for every clock pulse till the $V_{AD} \geq V_A$. The digital ramp ADC apart from the counter contains digital analog counter, analog comparator, and a control AND gate. The waveform at V_{AD} is a step-by-step process as it depends on the clock pulses and represents a staircase or ramp hence it is named as digital ramp ADC. The block diagram of a ramp type ADC is shown in Figure 14.4.

Working: The counter is reset to zero by applying the start pulse. The HIGH at the start stops the clock pulses from passing through AND gate into the counter. The DAC output will be zero ($V_{AD} = 0$) as all the inputs are zero. The comparator output will be high since $V_A \geq V_{AD}$. When there is a LOW at the start signal, the AND gates are enabled and clock pulses are passed on to the counter. These clock pulses fed to the counter are sent as binary digits to the DAC. The DAC will put out a series of staircase waveform rising step by step for each clock pulse and this is fed to the comparator as V_{AD}. The comparator has two inputs, the rising staircase voltage V_{AD} and the analog signal through sample-hold circuit. The output of the comparator is 1 when $V_A \geq V_{AD}$ and is fed to the AND gate. The other input for the AND gate is the

clock pulse. When both the output of the comparator and clock pulses are high, the AND gates enable the counter.

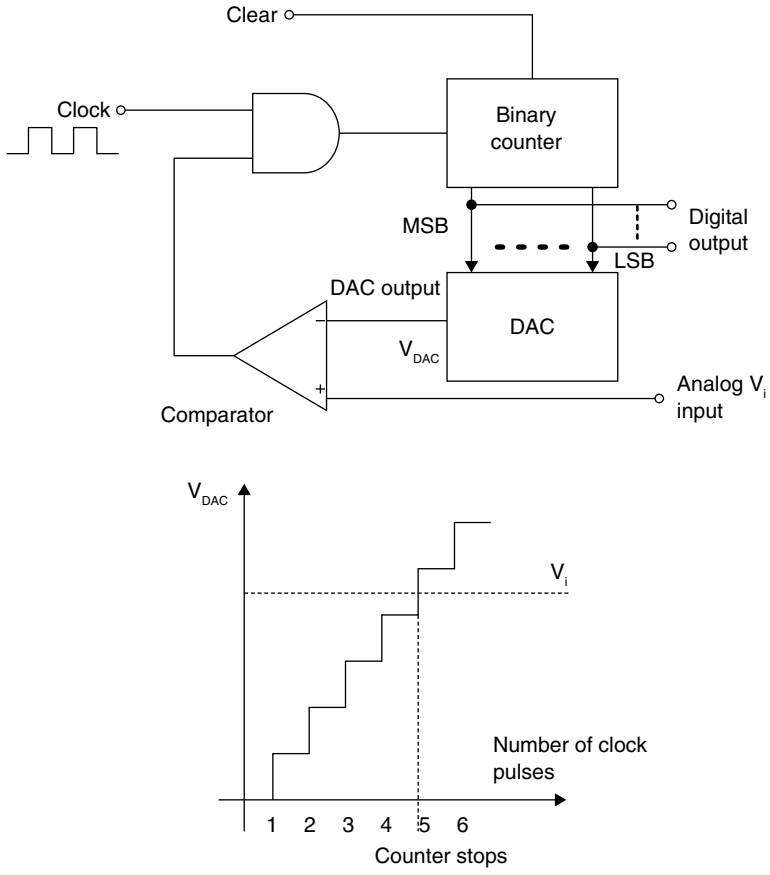


Fig.14.4 Block diagram and waveform of ramp type ADC

The counter counts and the binary number on the counter corresponds to the voltage of the input signal. The counter will hold this value till the next START pulse initiates a new conversion. The control circuit of the counter is provided with refinements like polarity check of the signal, positive or negative, overflow of signal in case the input exceeds the highest possible value of the DAC. The highest analog voltage that can be measured decides the output of the DAC and it depends on the counter. The digital ramp ADC is a slow convertor with the maximum voltage taking $2^n - 1$ clock pulses. The conversion time depends on the analog voltage V_A with the analog input voltage value. The ramp ADC resets itself after the completion of each measurement. The counter can be speeded up by replacing an UP/DN counter for $V_A > V_{AD}$, the counter counts up for $V_A > V_{AD}$ the counter counts downward. This is

known as "tracking ADC circuit". Since its circuiting is less complex it is used for low-speed application.

SUCCESSIVE APPROXIMATION ADC

Successive approximation convertor is the most widely used ADC. This convertor is more complex compared to the other types of ADCs and hence costlier. The conversion time is faster and is given as:

$$t_c = \frac{n}{f}$$

where n = number of bits

f = frequency

The conversion time is independent of the analog input voltage. The data can be drawn out of the ADC either in serial or parallel. The successive approximation type ADC uses a DAC in the feedback connection. The input is given to the comparator through a sample hold circuit. The sample hold circuit helps to maintain the input during the period of comparison. The important component of this ADC is the control logic component. The output from the control logic defines the successive steps of conversion. It is capable of modifying the data bit by bit. Once the START is given a high, then the control logic goes to high and the data stored in the register corresponds to the analog input V_A. The control logic is also connected to a ring counter. The counter provides the timing waveform of the clock pulses applied to it. These pulses control the operation of the convertor. During the first pulse from the ring counter, the D/A unit, n-bit register and ring counter all are reset. Thus, the ring counter has only the value 1 at MSB and all other are 0. The block diagram of successive approximation ADC is shown in Figure 14.5.

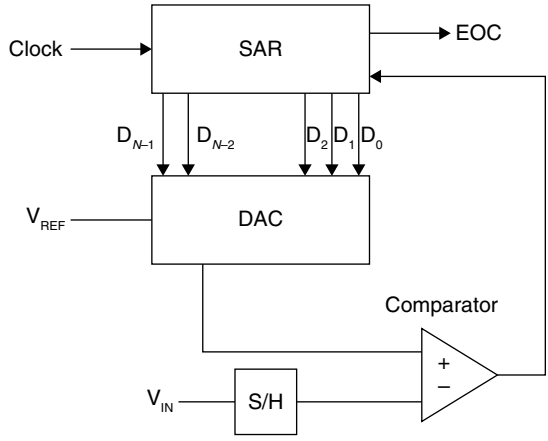


Fig.14.5 Block diagram of successive approximation ADC

Working: The operation of SAC is simple. The voltage of DAC that corresponds to MSD is compared by the comparator with the input voltage. If the voltage is less, the bit 1 is retained. If the voltage is more, it is reset to 0 and counter moves to next position. This process keeps continuing for each bit position until the nearest value is reached. The flowchart of the working of successive approximation ADC is given in Figure 14.6.

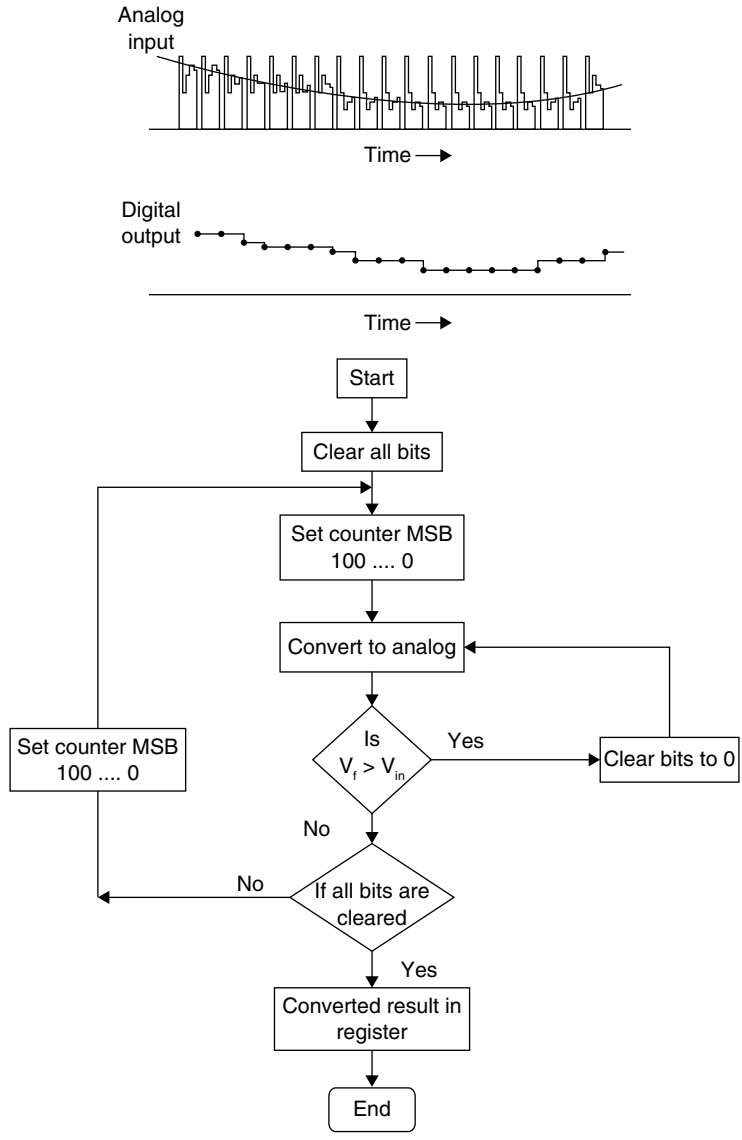


Fig.14.6 Waveform and flowchart of successive approximation ADC

Advantages

- The conversion time is comparatively faster. It is expressed as:

$$t_c = \frac{n}{f}$$

  where n = number of bits
  f = frequency
  The conversion time is not dependent on the analog input.
- It is highly reliable.
- It is more accurate than other ADCs.
- The output data can be taken out of the convertor either in series or in parallel.
- The sample hold circuit enables the input to be constant during the conversion process.

Disadvantages

- The circuit is more complex compared to ramp ADC.
- It is costly as it has complex circuitry.
- The speed of successive approximation ADC is limited to ~5 Msps.

Example 14.1 A counter type ADC contains a 3-bit binary ladder and a counter driven by 2 MHz clock. Calculate the conversion time and rate.

Solution:

$$\text{Conversion time} = \frac{2^n}{\text{Clock rate}} = \frac{2^3}{(2\times10^{-6})} = 44 \text{ sec}$$

$$\text{Conversion rate} = \frac{1}{t_c} = \frac{1}{(4\times10^{-6})} = 250 \text{ kilo conversions/sec}$$

Example 14.2 Calculate the conversion time of a SAC ADC which uses 2 MHz clock and a bit binary ladder containing 10 V register.

Solution:

$$T_c = \frac{n}{\text{Clock rate}} = \frac{4}{2\times10^6} = 2\ \mu\text{sec}$$

FLASH A/D CONVERTER

A flash ADC is constructed using a network of comparators and resistors. It is controlled by control logic. It has a decoder at the output terminal. For N-bit, a flash ADC comprises $2^N - 1$ comparators and 2^N resistors.

Working: Flash ADC uses the 2^N resistors to form a ladder voltage divider, which divides the reference voltage into 2^N equal intervals. The input voltage at a specific voltage range present among the 2^N voltage intervals can be found using $(2^N - 1)$ comparators. The output from the comparator is then fed to the combinational logic which then translates the information. This ADC does not require a clock so the conversion time is essentially set by the settling time of the comparators and the propagation time of the combinational logic. The circuit of a flash ADC is shown in Figure 14.7.

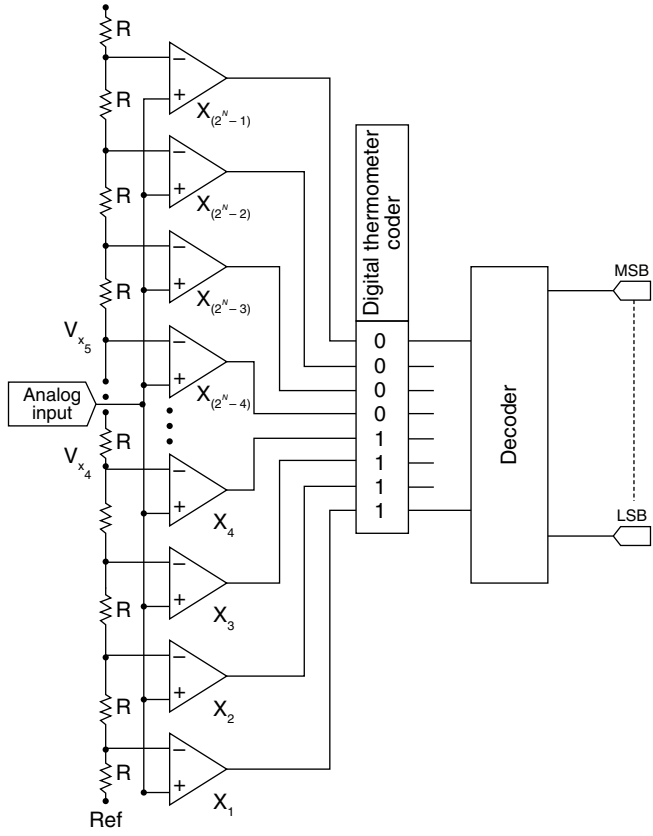


Fig.14.7 Flash ADC circuit

Advantages

- It is fastest ADC.
- The circuit is simple and has a simple operation.
- Since it has no clock, the speed is only limited by gate and comparator propagation delay.

Disadvantages

- It is expensive.
- Prone to produce glitches in the output.
- The number of comparators keeps increasing twice with each additional bit of resolution.

SIGMA–DELTA A/D CONVERTER

It consists of a simple circuit comprising of resistors, capacitor, comparators, control logic, and a DAC. The circuit is represented in Figure 14.8. The input analog signal is sampled and sent to the integrator. The output from the integrator is compared to ground.

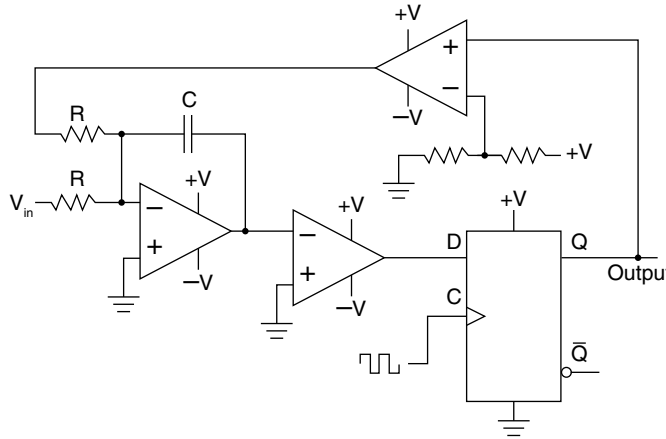


Fig.14.8 Sigma-Delta ADC circuit

This produces a serial bit stream. The output is a serial bit stream with the number of 1s proportional to V_{in}. This arrangement helps the sigma-delta modulator to adjust automatically to its output. Thus, it ensures that the average error at quantizing output is zero. The integrator value is the sum of all past values of the error, so whenever there is a non-zero error value the integrator value just keeps building until the error is once again forced to zero. The simplified circuit diagram can be represented as in Figure 14.9.

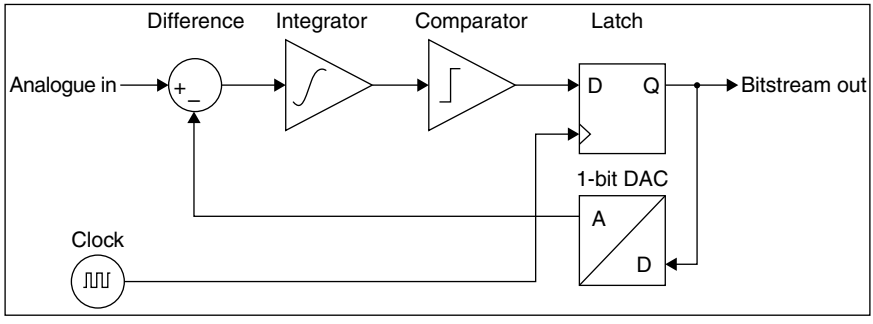


Fig.14.9 Simplified sigma-delta ADC circuit

Advantages

- High resolution.
- It does not require precision components.

Disadvantages

- Operation of the ADC is slow due to over sampling.
- Only good for low bandwidth.

DUAL SLOPE CONVERTOR

A dual slope ADC has a clock, counter and control logic as its components. It consists of an integrator circuit connected as feedback. The electronically controlled switches control the operation of the ADCs. It has a clock controlled counter that is fed through the comparator to the input terminal. The circuit of dual slope convertor is shown in Figure 14.10.

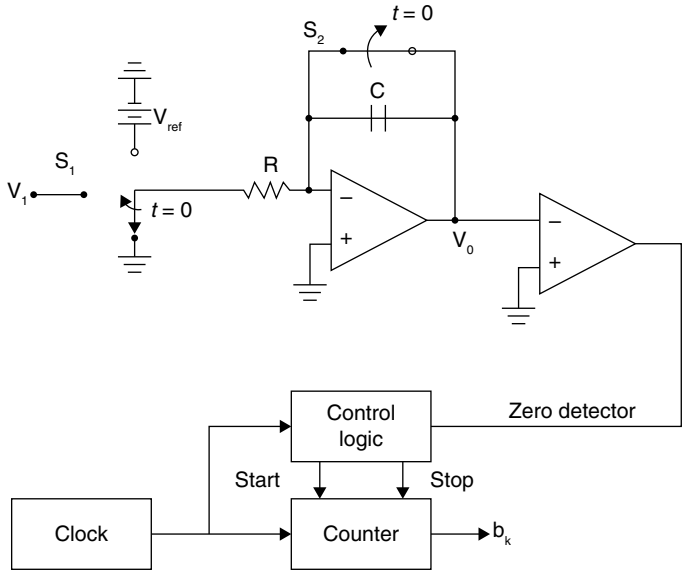


Fig.14.10 Dual slope convertor circuit

The given input voltage is integrated by passing it through the integrator circuit. It integrates the voltage for a fixed amount of time and later de-integrates by considering the reference voltage (V_{ref}). The errors formed during the process have no effect on the final conversion. This is because the errors formed are cancelled. At $t < 0$, S_1 is set to ground, S_2 is closed, and counter is equal to 0. At $t = 0$, a conversion begins and S_2 is open, and S_1 is set so the input to the integrator is V_{in}. S_1 is held for a constant predetermined time interval. When S_1 is set the counter begins to count the clock pulses, the counter resets to zero after predetermined time. The output of the integrator when $t = T_{INT}$

is given as $V_{out} = V_{in} T_{INT/RC}$. This is linearly proportional to V_{in}. At $t = T_{INT}$ S_1 is set so $-V_{ref}$ is the input to the integrator which has the voltage $V_{IN} T_{INT/RC}$ stored in it. The integrator voltage then drops linearly with a slope ($-V_{ref/RC}$). A comparator is used to determine the output voltage of the integrator when it crosses zero. When it is zero, the digitized output value is the state of the counter.

Advantages

- The final conversion result is insensitive to errors in the component values.
- It has only low adverse affects from noise.
- It has high accuracy.

Disadvantages

- The conversion process is slow.
- It requires precision components to improve accuracy.
- Since the components of the circuit are more, it is costlier.

DIGITAL-TO-ANALOG CONVERTORS

Conversion of digital-to-analog signals is very important for interfacing a digital system with analog system. They are used in the feedback of the ADC, the understanding of the DACs is required to deal with ADC. The input to be converted in case of DAC is usually in the form of bits 0 and 1 (binary digits). These are generally converted to voltage or current proportional to the digital value.

A general block diagram is shown in Figure 14.11.

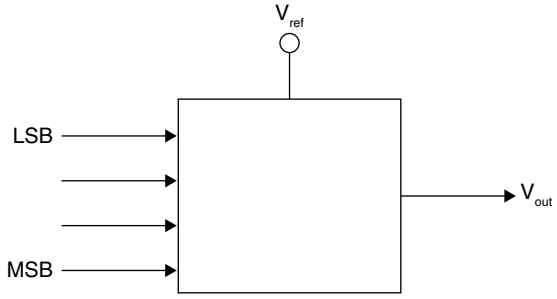


Fig.14.11 Block diagram of DAC

where a_0, a_1 … a_{n-2}, a_{n-1} is the word length for n bits. V_a is the reference voltage and the maximum analog output is

$$\text{Analog output} = k \times \text{digital input}$$

where k is the proportionality factor and is constant for a given DAC.

The DAC takes only specific values and not an analog quality. The possible output values are increased by the increase in number of input bits and the difference between

successive values can be decreased. The analog output for any digital input depends on the weight or the place value of the input binary data. The weight of each input gets doubled for each bit from LSB onwards.

Characteristics of DAC

- **Full scale voltage (FSV):** It is the maximum output that a DAC yields. This yield is obtained when all the bits of the digital input are 1. The output voltage corresponding to 1LSB for any bit is given as:

$$V_0 = \frac{V_R}{2^n}$$

where n is the number of bits.

If there are n bits, then the output is taken as the summation of outputs.

$$FSV = \frac{V_R}{2^1} + \frac{V_R}{2^2} + \frac{V_R}{2^3} + \ldots + \frac{V_R}{2^n}$$

$$= V_R\left(1 - \frac{1}{2^n}\right)$$

- **Resolution:** Defined as the smallest change that occurs in the analog output which is affected by a single step change in the digital input.

V = voltage difference of two logic values 0 and 1. This voltage is dependent on the number bits involved and hence the number of bits information is provided by the manufacturer as the resolution. The voltage improves as the bits increase. An 8-bit DAC has resolution of 0.39 and 12-bit DAC has 0.024. The percentage resolution of an N-bit DAC is given as

$$\% \text{ Resolution} = \frac{\text{Step size}}{FSV} \times 100$$

$$= \frac{1}{2^n - 1} \times 100$$

- **Accuracy:** This is defined as the error that arises between the actual outputs of DAC to the expected (or) theoretical output. It expressed as a percentage of FSV. If the convertor has an accuracy of ± 0.1 % for FSV of 10 V, then the minimum error for any output voltage is given as $10 \times 0.001 = 10$ mV. This in other words explains that the output of the given DAC deviates up to 10 mV from its theoretical value. The DAC's accuracy, is also contributed due to lack in linearity. Further limitations are contributed by uncertainly in reference voltage, amplifier gain, offset voltage, etc. For an ideal DAC, the accuracy and resolution are compatible.
- **Monotonicity:** Monotonicity refers to the change in the polarity. The output of DAC generally increases with the input voltage or decreases with the decrease in the input voltage. Sometimes the output of the DAC decreases with the increase in the input which causes a change in polarity. This is referred to as monotonicity and is to be avoided.

- **Linearity:** An ideal DAC has the analog output proportional to the digital input. Any increment in the input results in equal increment in the output. Thus, linearity gives a measure of the precision of a DAC. The standard linearity is ± ½ LSB. Linearity guarantees monotonicity and is dependent on the resistors and voltage drop across the switches. The resistors and switches are temperature dependent and this in turn affects the linearity. The temperature sensitivity ranges from about ± 50 ppm °C to as low as ± 1.5 ppm °C in a high quality unit.
- **Offset error:** An ideal DAC should yield a zero output when all inputs are zero. But this is not possible since there will be very small voltage called offset error. When the offset error is left uncorrected then it gets added to DAC output for all input applied. Many DACs are provided with an external offset adjustment that would enable to nullify the error.
- **Glitches:** These are spikes observed in the output due to the changes in the input when the transistors are turned ON or turned OFF.
- **(MDAC) Multiplying DAC:** This type of an option in the DAC enables to multiply the reference voltage and input. If the reference voltage is changed during the process of conversion, then alone it enables this option of DAC. When the reference voltage is changed then the output which is a product of reference voltage and digital input also changes.
- **Setting time:** This is the amount of time required for the conversion to yield a desired accuracy. The operating speed of a DAC is usually specified by its setting time and it is the time required to reach from 0 to full scale for the binary input charges from all 0 to all 1. The settling time varies from 50 ns to 10 μs. The settling time arises due to the fact that the stray capacitance and inductance need an appreciable time for any transitions and also the active devices like switches and transistors produce additional transients.

Types of DAC

- Binary Weighted Resistor
- Resistor Ladder DAC

Binary-Weighted Resistor DAC

The binary-weighted DAC consists of electrical components such as resistors for each bit of the DAC. These are connected to an adder circuit where the current across individual resistances are summed to the correct output value. Consider a 4-bit DAC using an output ampere performing summation which is connected using binary weighted resistors. The circuit diagram is shown in Figure 14.12.

The main components of a binary weighted resistance DAC are

- A weighted resistance network R to $2^{n-1}R$
- Switches equivalent to the number of bits applied

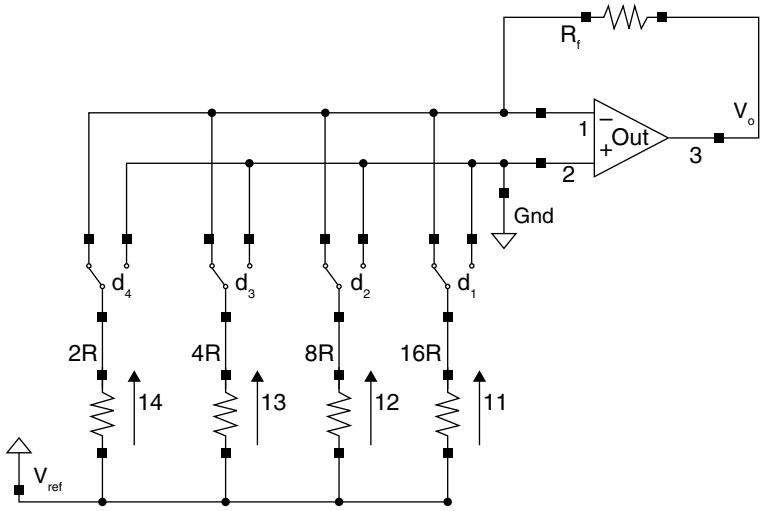


Fig.14.12 4-bit binary weighted resistor DAC

- Reference voltage V_{ref}
- The summing component that adds the current flowing through the resistors and gives a proportional output. It is usually an operational amplifier (OP-AMP).

Working: When a particular switch is closed, current flows through the connected resistor and the circuit through the resistors pass on to the summing element. The current produced when the switch connected to the LSB is closed is given as:

$$I\,(LSB) = \frac{V_{ref}}{(2^{n-1}R)}$$

This is also referred as the least current. The highest current produced when the switch connected to the MSB is closed and given as:

$$I\,(MSB) = \frac{V_{ref}}{R}$$

The input r_0 (LSB), Q_1,Q_2,Q_3 (MSB) are binary inputs that are assigned either 0 (logic 0) or 5V (logic 1) according to the input. The operational amplifier performs a weighted summation of the inputs. The op-amp works in the non-inverting mode and has a gain that is multiplied to each input voltage. The gain of the amplifier is given as:

$$V_{out} = \left(\frac{-R_f}{R_i}\right) V_{in}\ \text{amplifier}$$

The input resistors are binary weighted, i.e., starting from MSB resistor, the resistor value keeps increasing by a factor 2.

Therefore,
$$V_{out} = \left(\frac{-R_f}{R}\right)V_{Q_3} + \left(\frac{-R_f}{2R}\right)V_{Q_2} + \left(\frac{-R_f}{4R}\right)V_{Q_1} + \left(\frac{-R_f}{8R}\right)V_{Q_0}$$
$$= \frac{-R_f}{R}\left[V_{Q_3} + \frac{1}{2}V_{Q_2} + \frac{1}{4}V_{Q_1} + \frac{1}{8}V_{Q_0}\right]$$

The negative sign occurs as the input is given through the inverting terminal of the operational amplifier and is ignored in this case. If the feedback resistance R_f and the input resistors R is assumed to be equal, for e.g., if

$$R_f = R = 1\,\Omega \text{ then the output becomes}$$
$$V_{out} = -\left(V_{Q_3} + \frac{1}{2}V_{Q_2} + \frac{1}{4}V_{Q_1} + \frac{1}{8}V_{Q_0}\right)$$

It is evident from the equation that the voltage passing through Q_3 terminal is not attenuated, whereas the input voltage Q_2, Q_1, and Q_0 are attenuated by 1/2, 1/4, and 1/8, respectively. Therefore, the output ampere gives an output, i.e., weighted sum of all the digital inputs.

Example 14.1 Consider a digital input of 0101 then

$$V_{Q_3} = V_{Q_1} = 0 \text{ and } V_{Q_2} = V_{Q_0} = 1$$

Solution: Therefore,
$$V_{Q_2} = -\left[\frac{1}{2}(5) + \frac{1}{8}(5)\right]$$
$$= -\left[\frac{5}{2} + \frac{5}{8}\right]$$
$$= -[2.5 + 0.625]$$
$$= -3.125 \text{ V Output} = -1.875$$

For N-bit D/A convertor, equation be generalized

$$V_{out} = \frac{-R_f}{R\,2^{(N-1)}}\left[2^{N-1}V_{Q_{(N-1)}} + 2^{N-2}V_{Q_{(N-2)}} + \ldots + 2^1 V_{Q_1} + 2^0 V_{Q_0} - 2^{N-1}V_{Q_{N-1}} \ldots\right]$$

$$K = \frac{R_f}{R\,2^{N-1}}$$

Advantages

- This type of DAC is one of the fastest conversion methods.

Disadvantages

- Since the resistors are in the multiples of 2, as the number of bits keeps increasing the value of the resistance also increases.

- It has poor accuracy as it is difficult to obtain high precision at large values of resistance.
- Also these resistors are difficult to manufacture owing to their high values. This is the main disadvantage of the binary weighted type which is overcome by the resistor ladder DAC.
- Since high precision components are expensive, this DAC is limited to 8-bit resolution or less.

Resistor Ladder D/A Converter

The most reoccurred problem with binary weighted DAC is the high valued resistance between the LSB and MSB involved in the conversion of large number of bits. The fabrication of the resistors over a wide range without causing any change in the accuracy especially variations in temperature becomes highly meticulous. This can be solved by grouping the resistors in fours and applying a reference voltage which is 1/16 or 16 times larger.

In the R-2R ladder network, only two values of resistors are used. Each bit requires two resistors instead of one. The resistors that are used are close to each other in value. One of the most widely and simple type of DAC is the resistor ladder DAC. It comprises of operational amplifiers connected in the inverting mode performing as a summing network. A 4-bit R-2R ladder network having two resistor values R and 2R is depicted in Figure 14.13. The digital input is represented as Q_0, Q_1, Q_2, and Q_3 where Q_0 is the LSB and Q_3 is the MSB. Each bit in the circuit controls a switch between ground and the inverting input of the operational amplifier.

Let us consider the Thévenin voltage at MSB. All switches from S_0 to S_{n-2} are earthed. At the node of the switch S_{n-2}, Thévenin resistance will be 4R and the output voltage will be

$$\frac{V_R}{2} \times \frac{2R}{4R} = \frac{V_R}{4}$$

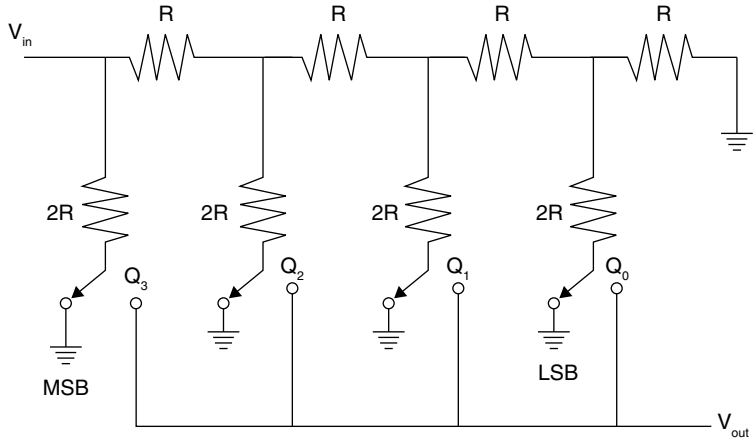


Fig.14.13 4-bit R-2R ladder network

At the 3rd node, Thévenin voltage will be $V_R/8$ in series with resistance R assuming the other switches to be earthed. If only three switches are present in the circuit, then the Thévenin voltage for MSB will be $V_R/2$ and $V_R/8$ for LSB.

Therefore, the output voltage $$V_{out} = V_R\left(\frac{1}{2^1}+\frac{1}{2^2}+\frac{1}{2^3}\right)$$

$$= V_R\left(\frac{S_2}{2^1}+\frac{S_1}{2^2}+\frac{S_0}{2^3}\right)$$

Multiplying and dividing by 2^n and putting the bit value a 0, $a_1,\ldots, a_{n-2}, a_{n-1}$ at the respective switches we get

$$V_{out} = V_R\left(\frac{a_{n-1}2^{n-1}+a_{n-2}2^{n-2}+\ldots+a_1 2^1+a_0 2^0}{2^n}\right)$$

$$V_0 = \frac{V_R}{2^n}\sum_{i=0}^{n-1} a_i 2^i$$

It could be better understood with an example, if the digital input is 0001 then the Q_0 switch is connected to the op-amp and all other switches are grounded. Since the non-inverting input of the op-amp is grounded, the inverting input is at the virtual ground. The resistance between node V_0 and ground R since it is parallel combination of two 2Rs. Therefore, V_0 is a result of the voltage division of V_1 across two series resistors of equal value R:

$$V_O = \frac{1}{2} V_1$$

Similarly, $$V_1 = \frac{1}{2} V_2 \text{ and } V_2 = \frac{1}{2} V_3$$

Therefore,

$$V_0 = \frac{1}{8} V_3 = \frac{1}{8} V_S$$

The gain of the amplifier is $-\frac{1}{2}$. Therefore, the output voltage corresponding to the binary input 0001 is

$$V_0 = -\frac{1}{16} V_S$$

Thus, the output of any combination of bits comprising the input binary number can be found by the principle of superposition

$$V_{out} = Q_3 V_{out3} + Q_2 V_{out2} + Q_1 V_{out1} + Q_0 V_{out0}$$

A negative reference voltage V_S can be applied to obtain a positive output voltage. Either a positive or negative range of output voltage can be obtained but not both. Therefore, only unipolar output and not bipolar voltage values are produced.

Example 14.2 A 4-bit R-2R ladder DAC has a reference voltage 4.5 V. It meets standard linearity. Calculate the percentage resolution, FSV, the output for 1010 and the range in output for 1010.

Solution:

(i) For R-2R ladder, percentage resolution $= [1/(2^n - 1)] \times 100$

$= [1/(2^4 - 1)] \times 100$

$= (1/15) \times 100$

$= 6.67\%$

Resolution in volts $= V_{ref}/2^n$

$= 4.5/2^4 = 4.5/16 = 0.2$ V

(ii) $FSV = V_{ref}(1 - (1/2^n)) = 4.5 \times (2^4 - 1)/2^4 = 4.2$ V

(iii) $V_{out} = V_{ref}(a_{n-1}\, 2^{n-1} + a_{n-2}\, 2^{n-2} + \ldots + a_1\, 2^1 + a_0\, 2^0)/2^n$

$= 4.5/2^4(2^3 + 2^1) = 4.5(10)/16 = 2.81$ V

(iv) Maximum deviation specific for linearity is ± 0.5LSB

LSB = Resolution in volts = 0.2V, therefore,

± 0.5LSB = 0.2/2 = 0.1 V

Range = 2.81 V ± 0.5LSB = 2.81 V ± 0.1 V

APPLICATIONS OF DAC

- It is used at the end of a digital processing chain where analog signals are required.
- Used in digital audio system like CD/MP3 players, HD radio, digital telephones.
- Digital video processing requires DACs. Example DVD players, DTV, and computer displays.
- They are used in industrial control systems like motor control, valves, transducer excitation.
- Used in waveform function generators and in calibration units in embedded systems.

MULTIPLE CHOICE QUESTIONS

1. A bio signal is converted to an electrical signal by passing it through a
   (a) Capacitor (b) Inductor
   (c) Transducer (d) Resistor
2. A 3-bit A/D convertor has _____ possible output codes.
   (a) 2 (b) 4
   (c) 6 (d) 8

3. The resolution of n-bit ADC with full scale voltage (FSV) is given as
   (a) $V = FSV/(2n - 1)$ (b) $V = FSV/(2^n - 1)$
   (c) $V = FSV/(2^n + 1)$ (d) $V = FSV/2^n$
4. The process of breaking down the continuous analog signal to discrete output states is called as
   (a) Quantization (b) Resolution
   (c) Accuracy (d) monotonicity
5. The conversion time for a counter type A/D convertor with 4-bit binary ladder and counter driven by 2MHz clock is
   (a) 4 msec (b) 2 msec
   (c) 8 msec (d) 1 msec
6. The conversion time of a n-bit successive approximation A/D convertor is given as
   (a) $T_c = 2n$/clock rate (b) $T_c = n$/clock rate
   (c) $T_c = (n - 1)$/clock rate (d) $T_c = n \times$ clock rate
7. A 'n'-bit flash AD convertor has _____ comparators and _____ resistors.
   (a) $2n - 1$ and 2^n (b) $2n + 1$ and $2n$
   (c) $2n$ and 2^n (d) $n - 1$ and n
8. An 'n'-bit digital ramp ADC, the maximum voltage takes _____ clock pulses.
   (a) $2n + 1$ (b) $2n - 1$
   (c) $2^n - 1$ (d) $2n$
9. The number of comparators used for flash ADC keeps _____ with is each additional bit of resolution.
   (a) Decreasing (b) Increasing
   (c) Increasing twice (d) Decreasing twice
10. The percentage resolution of a 5-bit DAC is
   (a) 3.22% (b) 6.67%
   (c) 9.67% (d) 12.9%
11. An 'N'-bit weighted resistor DAC requires _____ precision resistors.
   (a) $2N$ (b) N
   (c) $N + 1$ (d) $N - 1$
12. A 3-bit R/2R digital-to-analog (DAC) converter has a reference of 5V. What is the analog output for the input code 101?
   (a) 1V (b) 0V
   (c) 0.625 V (d) –6.25 V
13. In a digital to analog converter, the smallest change in the analog output due to the increase in the digital input is known as
   (a) Quantization (b) Resolution
   (c) Accurracy (d) Linearity

14. A 4-bit DAC has reference voltage of 5V then the full scale voltage is
    (a) 4.68V (b) 4 V
    (c) 5 V (d) 5.4V
15. Which of the following is a disadvantage of flash ADC?
    (a) Long conversion time
    (b) Large output lines
    (c) Input voltage is applied simultaneously
    (d) Large number of comparators
16. During the conversion of analog to digital process, the input analog signal is stabilized using
    (a) Encoders (b) Transducers
    (c) Sample hold circuits (d) None of the above
17. Which of the following ADC has the maximum conversion speed?
    (a) Counter type ADC (b) Dual slope ADC
    (c) Digital ramp ADC (d) Successive approximation ADC
18. If a binary weighted DAC has a feedback resistance of 10KΩ and current of 25μA, then the output voltage is
    (a) 0.25V (b) –0.25V
    (c) 25V (d) 250V

Answers

1. (c) 2. (d) 3. (b) 4. A 5. (c) 6. (b) 7. (a) 8. (c) 9. (c)
10. (a) 11. (b) 12. (c) 13. (b) 14. (a) 15. (d) 16. (b) 17. (d) 18. (b)

DESCRIPTIVE QUESTIONS

1. Explain with a neat diagram, the working of digital ramp A/D convertor.
2. Explain the working of a 4-bit R-2R binary ladder D/A converter.
3. Draw and explain the working of a successive approximation A/D converter.
4. Write notes on the following characteristic of the operation of an ADC
   (a) Quantization
   (b) Sample and hold circuit
   (c) Resolution
5. What are the specifications/characteristics used by the manufacturers to describe a digital to analog converter? Explain each one briefly.
6. Draw and explain the function of dual slope analogue to digital converter. Derive the equations used.
7. Determine the analog output voltage of 8-bit DAC (R-2R ladder network) with V_{ref} as 5V when the digital input is 10111001.

8. With the help of a neat diagram, explain the working of a weighted-resistor D/A converter.
9. Write note on the conversion time of successive approximation ADC.
10. Mention the advantages of the flash type of ADC over the successive approximation ADC.
11. Draw a diagram to illustrate the structure of a flash analogue to digital converter and briefly explain the principle of operation of the device.
12. Write short notes on the following characteristics of a DAC
    (a) Resolution
    (b) Monotonicity
    (c) Linearity
    (d) Settling time

COMPUTER FUNDAMENTALS

INTRODUCTION

Computers have become an integral part of our lives. The world today completely depends on computational technology. People in olden days used their fingers and pebbles for computation. The word "digital" is derived from Latin and is interpreted as fingers, and calculus refers to pebbles. Computer originated from the term "compute" meaning calculate. The advent of computers has minimized the tedious calculations and has shrunk the world connecting peoples across the poles of the earth.

Computers are electronic devices that perform calculations and controlling operations that are expressed in logic or numeric terms. Computers are broadly classified into analog, digital and hybrid computers. Analog computers deal with continuous quantities such as temperature, pressure, current, etc., whereas digital computers operate on digital signals. Hybrid computers, on the other hand, are designed to perform both analog and digital functions.

COMPUTER CHARACTERISTICS

Computers possess certain characteristics that enable them to execute logical operations, data processing, etc. These characteristics form an essential part for emerging technologies.

- **Speed and accuracy:** The prime function of computers is high speed data processing. It reads millions of instructions every second and executes them; therefore, computers are required to possess great speed. Generally, the speed of computers is expressed in terms of GHz; but recent computers have even greater speed ranging from billions of operations per second.
- **Diligence**: The computer, irrespective of the number of calculations to be performed, must have the same speed and accuracy for all operations. The speed and accuracy of the computer must remain stable from the first until the last operation.

- **Reliability:** The performance of computer measured against predetermined operation metrics is evaluated as its reliability. This is an important character. The hardware of the computer. Yet the computer has inbuilt diagnostic capabilities that monitor the system throughout.
- **Storage competence:** The storage capacity of computers is high. Data can be stored and retrieved instantly. A part of the data is stored in the memory devices; however, the storage capacity is augmented by storing the remaining data in secondary memory devices such as disks. Thus, the data can be accessed and processed when required.
- **Adaptability and resource sharing:** Computers are versatile and can multitask. The user can perform any number of operations at a given time with the help of various computer programs. This makes the computers useful in resource sharing where external devices like printers can be linked to a number of computers and operated with a set of instructions from any computer. Stored data can be transferred from one system to another thereby creating a larger data base.

Although computers have all the above characteristics, they still have a few limitations. Computers can only perform instructions that are pre-programmed. The instructions must be clear and well defined to avoid any confusion. The usage of computers is limited to qualitative considerations. Certain precaution for the maintenance of computers is essential. They have to be placed in dust-free zone. Heavy processing generates heat in the intricate parts of the computer, therefore, a low ambient temperature is to be maintained.

EVOLUTION OF COMPUTERS

Computers originated when there was a need for devices that could calculate and do other operations in minimum time. Before the advent of computers, people used their fingers and pebbles to perform basic mathematical operations. In the olden days, people used a counting device called “abacus” that consists of beads that slide along parallel fixed rods. It permitted users to perform basic calculations; and due to its simple construction, it is popular in many countries even today. In 1614, Napier, a Scottish mathematician, developed the computing methods later named as Napier Bones. A few centuries later, the next significant advancement in computers arrived with the invention of automatic calculator by French philosopher and mathematician Blaise Pascal. The instrument consisted of gears, dials and wheels that could be operated mechanically to perform calculations.

German mathematician Gottfried Wilhem von Leibnitz later brought in improvizations to Pascal’s calculator thereby enabling the device to determine multiplication, division and square root. The only drawback was the reliability of the device as it lacked precision due to its construction. The transition to computers was however, made by Charles Babbage, the mathematician at Cambridge who developed a device that could store and read information. He proposed a device called the difference engine to perform differential equations. He made further modifications that resulted in

the analytical engine. The input devices included perforated cards containing operating instructions and a storage memory that could hold 1000 numbers up to 50 decimal digits long. It contained a control unit that allowed random sequencing of processing instructions. The output device produced printed results. Due to inadequate technologies, the analytical engine was never constructed but laid the foundation to basic elements of modern computers. The pioneering efforts contributed by Charles Babbage made him the father of computers. Herman Hollith, a statistician at the US census bureau devised cards to store data information which he fed into a machine that compiled results mechanically. In 1889, he used these punch cards to complete census that took just two and half years. His company "The Computing Tabulating Recording Company" was later renamed as IBM—International Business Machine Corporation. Harvard Aiken and IBM in 1944 invented the electronic calculating machine that comprises of relays and electromagnetic components that replaced mechanical components.

In 1946, John Echert and John Mauchly of Moore School of Engineering developed ENIAC (Electronic Numerical Integrator and Calculator). This computer used electronic vacuum tubes that form the internal parts of the computer. Further developments on this computer led to EDVAC (Electronic Discrete Variable Automatic Computer). In 1951, UNIVAC (universal automatic computer) was manufactured and this marked the real beginning of the computer era.

By 1960s, computers with greater capacity and faster operating speeds were developed. The manufacturing of such computers increased with a wide range of capabilities and prices, accessories such as CRT displays, printers, etc., were also provided. Then, transistors became the norm and entire computers were made available on tiny chips known as integrated chips. By the 1980s, VLSI design was introduced on which millions of transistors were placed on a single chip and the size of the computers shrunk. The personal computers came into the scene which were user friendly and cost effective.

Microcomputers equipped with read only memory (ROM) performed many functions and could be used to store data constantly. During the late 1980s, PCs were run by microprocessors that were capable of handling 32-bit data at a given time and processed 4,000,000 instructions per second. These transformations resulted in the invention of microcomputers that wore mini processors on a single integrated chip (IC).

The computer industry has grown in leaps and bounds in the recent years with the introduction of electronic mail, electronic publishing, networking, etc. Advancements in the technologies have invented cheaper and powerful computers.

GENERATIONS OF COMPUTERS

The history of computers has been classified with reference to the generation of computers. Generation here refers to the period of technological development or invention. The technological advancements made computers more powerful, reliable, improved operating speed, smaller and cost effective. Based on the processor installed, computers are classified into five generations as discussed in the following sections.

First Generation (1940–1956): Vacuum Tubes

The first generation computers were built using vacuum tubes or thermionic valve-based machines. A cylinder coated with iron oxide called magnetic drum formed the storage element. The magnetic drums were used for memory and the circuits used vacuum tubes. The inputs were provided through punch cards and outputs were displayed in the form of printouts. They had comparatively faster computing that extended to milliseconds. First generation computes relied on binary coded language for all operations, but were able to solve only a single operation at a time. They lacked versatility and speed. Examples of first generation computers are ENIAC, EDVAC and UNIVAC.

Characteristics of First Generation Computers

- The first generation computers were based on vacuum tube technology.
- They were larger in size and occupy more space.
- They were not handy and slow.
- They lacked adaptability and speed.
- They were not reliable and face hardware failures.

Second Generation (1956–1963): Transistors

Second generation computers used transistors that had more superior performance than vacuum tubes. The transistor is a small device and hence the size of the computers could be reduced greatly. Computers became faster, smaller, cheaper, more energy efficient and much more reliable than the first generation computers. Since transistors offer less power consumption, these computers had low power requirements. The primary memory was the magnetic core and the secondary memory devices were the magnetic disks. Although the memory devices were installed, these computers still relied on punch cards for the input, and output was in the form of printouts. The second generation computers paved way for the assembly language. These computers began to operate on assembly language instead of machine language. Mnemonics were used to express the assembly language as an alternative for numbers. SUB was used for subtraction and MULT for multiplication. This made the programming much simpler. High level programming languages like COBOL and FORTRAN were introduced during this phase. Some of the examples for second generation computers are IBM 1401 and IBM 7090.

Characteristics of Second Generation Computers

- Transistor technology was used for the manufacturing of these devices.
- The size of the computer was reduced drastically compared to its first generation counterpart.
- Since the size was smaller, they were portable.
- Assembly language came into existence thereby resulting in much efficient programming.

- The operating time was reduced from millisecond to microseconds.
- Less heat generation during operation.
- These computers were more reliable as they did not face hardware failures.
- Ambient temperature was required.

Third Generation (1964–1970s): Integrated Circuits

Third generation computers were formed using integrated circuits. Integrated chips abbreviated as ICs consist of a single chip comprising of many transistors and resistors. The complex circuits were replaced by a single IC chip, thereby reducing the size drastically. The introduction of ICs improved the reliability and efficacy of these computers. The input and output units like punch cards and printouts were replaced by keyboards and monitors. These devices were interfaced with operating system that permitted the computer to perform many applications at one time with the main program governing the memory. Since a single chip replaced intricate components, the cost of the system was greatly reduced. The examples for third generation computer are NCR 395 and B6500.

Characteristics of Third Generation Computers

- Integrated technology was used to design the third generation computers.
- The computational time was minimized from microseconds to nanoseconds.
- The reliability was greatly improved compared to second generation computers.
- The reduction in size made them easy to handle and portable.
- The operating speed was very fast and generated less heat.
- The power consumption was less.
- Mass production was easier and cheaper.
- Hardware failures hardly occurred yet the manufacturing of ICs was more intense and required high standard technologies.

Fourth Generation (1970s onwards): Microprocessors

These computers were built using the microprocessors that contain millions of transistors. Although the integrated chips were still in use, the microprocessors took them one step ahead. In 1971, Intel 4004 chip was developed that had the CPU, input–output controls and memory on a tiny chip. Microprocessor chip is a silicon piece of about 0.5cm thick. Large scale integration (LSI) and very large scale integration (VLSI) was introduced with the fourth generation computers. These computers were more reliable, potent, compressed and cost effective than the previously invented computers. The semiconductor memories replaced the magnetic core memories. The cost of secondary memory devices were cut down to a greater extent. The capacity of the memory devices were increased manifold but the size reduced considerably. This generation saw the development of devices such as mouse and GUI (graphical user interface). Apple II, Altair 8800 and CRAY–I are the examples for fourth generation computers.

Characteristics of Fourth Generation Computers

- The microprocessors formed the basis for the design of fourth generation computers.
- The fourth generation computers are compact and the cheapest among all other generation computers.
- They are extremely portable.
- Heat generated is negligible and thus the requirement for ambient temperature is not needed.
- They had better communication and interfacing.

Fifth Generation (Present and Future): Artificial Intelligence

These computers are created to perform decision making and reasoning like humans. If the computer is programed by a series of command like "if", "when", "then", it analyses the entire function and takes decision. It was in the 1990s that these computers came into existence. The future would use the super large scale integrated chips (SLSI) that would accommodate millions of electronic components on a single microprocessor chip. Since the instruction set is more, larger memory space is required to store information, therefore, mega chips may enhance the storage. These computers lack multitasking ability and they are designed with the aim of troubleshooting the drawbacks of parallel processing. The characteristic feature of this computer is its ability to perform several instructions at the same time through numerous CPUs. Artificial intelligence refers to a series of technologies that could be similar to human behaviour. These technologies can reason and decide like humans at a given situation. Among the many subclasses of AI, robotics is a well-known branch that is gaining enormous popularity.

CLASSIFICATION OF COMPUTERS

Computers come in various specifications. They are classified based on their hardware, size, utility and capacity. Based on the above parameters, the computers are classified as follows:

Data Handled Type

Based on the data handled by the computers they are classified as follows:

- Analog
- Digital
- Hybrid

Analog Computers: Analog computers are used for measuring any two variables by continuously applying electric signals. These computers operate by applying electrical pulses. They are used to measure quantity of any continuous physical phenomena such as electrical, mechanical or hydraulic quantities and process them into desired output. The display in the case of analog computers is continuous and the output is displayed in graphical format plotted between the two measured variables. The electrical input

signal is converted into non-electrical data. Generally, the measured variables are physical quantities such as temperature, pressure, current, voltage, etc.

Characteristics of Analog Computers: The characteristics of analog computers are as follows:

1. They operate by measuring analog signals.
2. They have poor accuracy.
3. They have inadequate memory space for storage purposes.
4. They are restricted only to limited applications that involve the measurement of physical quantities.
5. The operating speed is very low as it depends on electrical signals.
6. They are not suitable for commercial purpose.

Digital Computers: Digital computers operate using digital values. They operate with both numeric and alphanumeric values. Unlike analog signals, here the digital input is in discrete form and operates on counting the input data. Digital signals are represented as binary data 0's and 1's. The 1 in binary form represents ON state and 0 represents an OFF state. They can perform mathematical operations such as addition, subtraction, multiplication, division and exponents. However, addition is carried out first and later followed by other operations. Since they operate with discrete voltage signals, they are much faster and more precise than analog computers. Digital computers have been extensively used for commercial applications. They perform low speed sequential as well as parallel processing. The output is continuous but is obtained only on the completion of the computation.

Characteristics of Digital Computers: The characteristics of digital computers are as follows:

1. The given input data is converted to binary form as all calculations are in binary system.
2. They operate with discrete numbers.
3. They are much more accurate and faster.
4. They are versatile and hence have wide range of applications.
5. The memory space is large and therefore processing speed is high.

Hybrid Computers: Hybrid computers were created with the objective of combining both analog and digital data processing. The measured quantity in case of a hybrid computer is analog while the processed output is in digital form. These are used when the input measured is a continuously variable physical quantity such as blood pressure, temperature, etc., and after being processed gives a digital output or graphical output. Hybrid computers are used in medical fields in measuring biomedical parameters, weather monitoring units, temperature monitors, etc.

Based on Utility

According to purpose of utility, computers are classified into general and special.

General Purpose Computers: General purpose computers are designed to perform a wide range of tasks. They have the capability to store copious programs and information, however, the operating speed is low. The efficacy of these computers is less than special purpose computers. These computers can be used in both commercial and non-commercial sectors. Since they are used in many diverse applications they have remarkable versatility. Most of the computers now are general purpose digital computers.

Specific Purpose Computers: Special purpose computers are designed to handle a specific assignment or to perform a specific task. A set of instructions suitable for the desired task is built into the machine. The versatility is restricted such that only specific tasks can be performed by specific computers. These computers are required to possess great accuracy and response.

Based on Size and Capacity

According to the size and storage capacity, computers are classified as follows:

- Microcomputer
- Mini computer
- Mainframe computer
- Super computer

Microcomputers: A microcomputer is a petite and a relatively inexpensive computer with a microprocessor as its central processing unit (CPU). The microprocessor is an integrated chip that has components such as input output device and CPU fabricated on it. The advent of efficient microprocessor initiated the development of these computers in the 1970s and1980s. As the name suggests, these computers were of relatively small size compared to mainframe computers and minicomputers. These computers are single user systems that are designed to operate basic functions. They are the most common category of computers and are also known as personal computers. For example, IBM PCs, Apple Mac, etc.

Advantages of Microcomputers

1. These computers are physically small and hence easily transportable.
2. They occupy very less space.
3. They consume less power.

Disadvantages of Microcomputers

1. They are relatively slow for random access.
2. They have very less storage capacity.

Minicomputers: Minicomputers are larger than microcomputers and have more processing power. They are also known as mid range computers. Minicomputers are generally multiuser computers that possess multiprocessor system. They have more memory capacity and word length of 32-bits. The processing speed is around

10–30 MIPS (million instructions per second) and has a RAM capacity of 8 MB–256 MB. Microcomputers are used in workstations that require high performance and graphical input/output information.

Advantages of Minicomputers

1. It is a general purpose computer that is compatible for high level programming languages.
2. It has a word length of 32-bits.
3. It can perform the functions of a mainframe computer and yet is inexpensive.

Disadvantages of Minicomputers

1. The operating speed is less.
2. The storage capacity of minicomputers is less compared to mainframe computers.

Mainframe Computers: The term "mainframe computers" were intended to formulate a difference from the traditional minicomputers. Mainframe computer is faster, more expeditious and more expensive than other general purpose computers. They are acclimated to handle immense volumes of data. The word length may be 48–64-bit recollection capacity, the hard disk capacity may be 1000MB–10 GB. They are measured in MIPS (million instructions per second) and respond to up to 100s of millions of users at a time. Since these computers are capable of handling and processing very large amounts of data quickly, they require highly efficient operating systems. They are utilized where a substantial amount of data is to be processed and very intricate calculations are to be made. These tasks are beyond the computing capacity of minicomputers. For example, IBM 308X, IBM 3090 series, HP 9000, etc.

Supercomputers: Supercomputers are the most powerful of all computers. They have a high processing speed that ranges from 400 to 10,000 MIPS and word length of 64–96-bit. It has a memory space of 256MB, hard disk capacity 1000MB or more and machine cycle time of 4–6 ns. PARAM is one of the best developed supercomputers that was designed in India. As the name suggest, they can perform billions of instructions per second. The latest supercomputers can be equated to 40,000 micro computers.

Advantages of Supercomputers

1. Multiple tasking and parallel processing is performed simultaneously.
2. The operating speed is swift and has more memory space.

Based on Usage Mode

Computers can be classified based on the usage mode as palmtops, laptops, personal computers, work stations, game consoles, notebooks, tablet smart books, etc. Technological improvizations are being made every day that give rise to more efficient and compact computer systems.

Palmtops: Palmtop computers are as small as the size of the palm. The input is handwritten on the screen using a special purpose electronic pen. Although the storage capacity is less, the input is handwritten and it is easier to operate. The user needs to be trained in its operation. A palmtop has a phone, fax and e-mail options embedded in it. For example, windows CE.

Laptops: These are easily portable computers that have inbuilt memory storage and are also compatible to external storage devices. The functions and arrangement of laptop is very similar to a computer and looks more like its miniature. Although it is more advantageous than a palmtop, it also has few disadvantages such as short battery life, mini laptops or notebooks lack CD drive, etc. They are more expensive than table top computers.

Personal Computers: These computers are intended for personal use and are larger in size compared to laptops and palmtops. They have excellent storage capacity, however, since they are heavy they are non-transportable. The configuration of these computers keeps varying depending on the manufacturers. They are generally available in two models—desktop model and tower model.

Workstations: These computers are desktops intended for multiprocessing facilities. Their storage capacity is high. They are similar to personal computers but are all connected and controlled by the local Internet network.

Servers and Clients: When numerous computers are involved in a computation it is practically not possible to control them individually hence these computers are interconnected over a network. This enables sharing of information uniformly to multiple users.

Server: A server is a larger computer that facilitates sharing of resources and services. These provide shared resources to the client.

Client: It is a single user computer that interfaces with the end user. It is used for communicating with the server.

COMPONENTS OF DIGITAL COMPUTERS

Computers process data electronically with less or no human interference. There are several computer systems with various models, brands and peripherals. A digital computer mainly consists of five components. They are the input device, arithmetic logic unit, control unit, memory unit and output devices, and storage unit. A block diagram of computer architecture is shown in Figure 15.1.

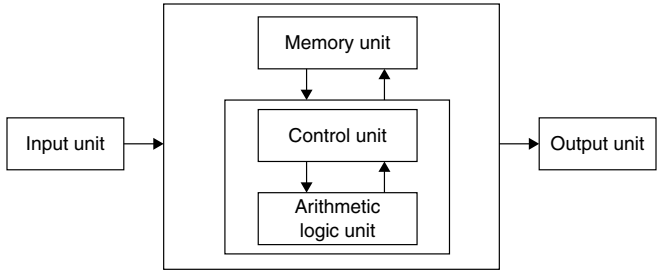


Fig.15.1 Block diagram of computer architecture

Central Processing Unit (CPU)

The central processing unit (CPU) is considered the brains of the computer. It executes command, interprets operations and controls the instructions. The CPU is binary coded, i.e. works only with 0's and 1's. After performing the desired action, the result is converted to alphanumeric and interpreted by the user. The CPU is built on a microprocessor. The CPU has registers as its storage memory. The data is stored till the next operation is directed by the control unit. The CPU has three sub units:

Arithmetic Logic Unit (ALU)

Arithmetic and logic operations are performed on the given data. Arithmetic operations such as addition, subtraction, multiplication and division are performed swiftly on the data by the arithmetic unit. The logic unit does logical comparisons between data. The three main logical operations performed by the logical unit are equal to, less than and greater than. These conditions are compared between two values. The data stored in memory is transferred to ALU, and on completion of the desired operation is again stored in the internal memory registers. The data is transferrable between the ALU and internal memory provided the operation is not completed. Once the operation is finished it is transferred from internal storage to output device.

Control Unit (CU)

The sequenced data from the ALU enters the control unit. The control unit checks for the occurrence of errors in the order of the programs executed. Program instructions are interpreted and checked for correctness. Electrical signals are used to control, execute and store instructions. It only directs the instruction and does not perform it. To operate the control unit there exist few registers called instruction registers. Instruction register holds the instruction and instruction pointer holds the successive instruction. The instructions are decoded so as to be executed; the operations of the control unit depend on the instruction. It also controls the input output devices and the overall functioning of the computer.

Memory Unit (MU)

This is the main memory and forms the integral part of the CPU. The primary memory is of two types, namely, random access memory (RAM) and read only memory (ROM).

RAM provides information to the processor. It comprises of sequential memory that has unique address and contains a data element. The information stored in RAM is erased once the power is switched off and hence called volatile. RAM has been dealt elaborately in the earlier chapters. ROM is used for storing start up instructions and routines in basic input/output system. Unlike RAM, these are non-volatile devices. The types of ROM are already described in earlier chapters.

Cache Memory

Cache is a small high-speed memory usually static RAM (SRAM). It stores data from some frequently used addresses of main memory. It is also used in speeding up the

memory retrieval process. High speed is essential since the processor takes a longer time to bring the instruction then to execute it. The access time for DRAM is 60ns. A 100 MHz processor can execute most instructions in 10 ns. Cache memory helps by decreasing the time it takes to move information to and from the processor. Some of the common terms used when talking about cache are discussed as follows.

- **Cache hits:** When the cache contains the information requested, the transaction is a cache hit.
- **Cache miss:** When the cache does not contain the information requested, the transaction is a cache miss.
- **Snoop:** When a cache is watching the address lines for transaction, it is called a snoop. This function allows the cache to see if any transactions are accessing memory it contains within itself.
- **Snarf:** When a cache takes the information from the data lines, the cache is said to have snarfed the data. This function allows the cache to be updated and maintain consistency. Snoop and snarf are the mechanisms that the cache uses to maintain consistency.
- **Dirty data:** When data is modified within cache but not modified in main memory, the data in the cache is called "dirty data."
- **Stale data:** When data is modified within main memory but not modified in cache, the data in the cache is called "stale data."

Secondary Storage Devices

The primary memory is the main memory of the computer. The storage exceeds the capacity of the main memory and hence secondary storage devices are required to support a computer. The secondary storage systems are of various forms such as disk memories, magnetic tapes, optical disks, compact disks, etc.

Types of Disks

Earlier disks were hard magnetic disks. Types of magnetic disks have evolved over the years. The types of disks are flexible/floppy disk and Winchester disks. The use of optical property to store or read information led to CD-ROMs and optical devices. Laser is used to read data in CD-ROM disks, while the hard disk, floppy disks and Winchester disks use magnetic head read or write data. The disks rotate on a unit called driver. CD-ROM is written once and can be read as many times by the users. The rewritable disks are commercially available at affordable cost. Floppy disks have become extinct with the advent of pen drives and external memory disks.

Magnetic Disks: Magnetic disk consists of multiple circular aluminium disks called platters. The disks have a common axis fixed in a common shaft. A gap is present between the consecutive disks. The rotating heads move in gaps. The disks are coated with cobalt and nickel. When current is passed through the coil in the recording head, the information is recorded. This creates magnetic fields and can be viewed as magnetic

bands. These bands are magnetized by controlling the current thereby recording 1 or 0-bit. When the next bit is to be stored, the polarity of one band is maintained opposite to the polarity of the band preceding it. This indicates the beginning of the next bit. To read the data, the flow of current is restricted in the coil. The polarity of the current determines the read bit 1 or 0.

Hard Disks: The hard disk consists of a flat surfaced platter called the slide. On each side of the platter, a cylinder is placed for a set of tracks. The cylinder is at equidistance from the common axis that passes through the centre of the disk. Each platter has two disks, therefore, if a disk has three platters then each cylinder will have six tracks as is shown in Figure 15.2.

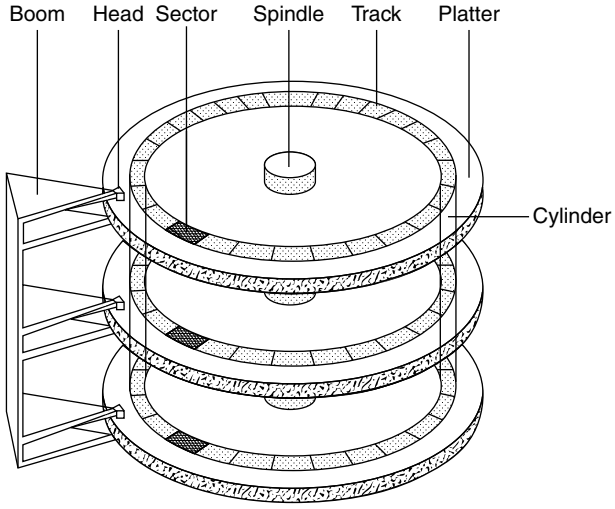


Fig.15.2 Cylinders of hard disk

Winchester Disks: Winchester disk indicates a wide range of small disk drives. It is sealed in a chassis to prevent accumulation of dust. The head of the disk is mounted on a holder which is positioned at about 0.5–2.5 microns from the platter surface. During the stationary mode, the surface of the disks rest on the head for landing and the area is called landing area. The data is not stored at this position. The read and write heads are mounted on the actuator arms pivoted on the shaft. They can be moved with the help of a rotator head positioner. The arm is rotated around the shaft by the voice coil actuator. Winchester technology was originally applied to disks of size 35cm and was later extended to disks of smaller size.

Floppy or Flexible Disks: The floppy disks are simpler secondary storage device. These disks are made of a sheet of plastic that are easily transportable. A typical floppy disk is illustrated in Figure 15.3. It consists of a hole in the centre called the index hole. When the index hole passes between optical source and a photoelectric cell, it helps to detect the starting point of a track. The head access hole is provided to enable

the read/write head to contact the desired location on the disk. The early floppy disks invented by IBM measured 8 inch diameter; but later with the invention of personal computers, the size of the floppy disks reduced to 5.25. Initially, the disks were enclosed in cardboards. The disk also consists of a metal hub in the centre and the head access is covered by a sliding metallic cover. At the corner of the disk there are two notches. The first notch is provided only in high storage disks that can hold information up to 1.44MB while the second notch is provided for both low (720KB) and high capacity disks. By sliding the cover the second notch is open. No information can be written to the disk when the second notch is opening, therefore, it called write protect notch. The read/write head of the floppy disk is moved to the desired track by a stepping mechanism.

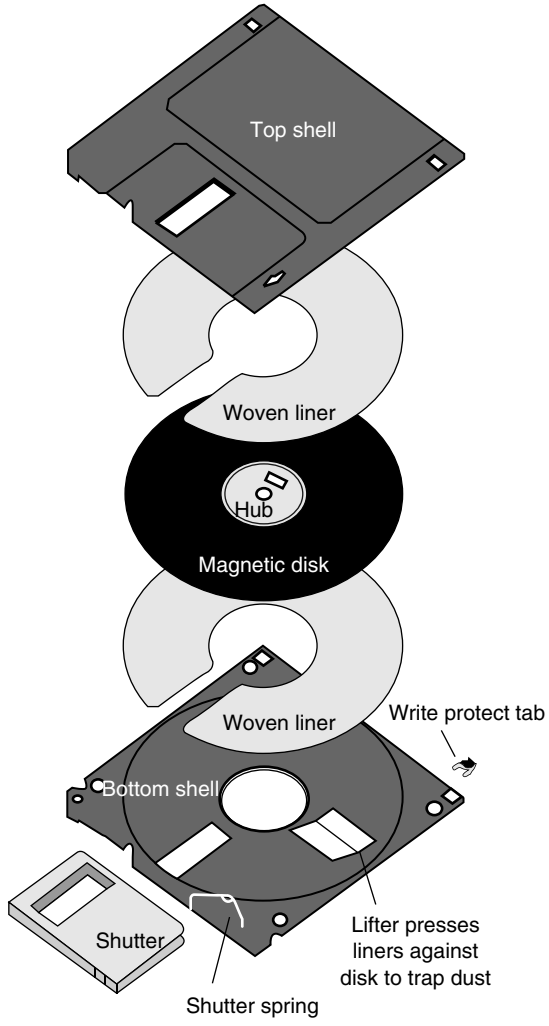


Fig.15.3 Typical floppy disk

The head is positioned over the disk and accesses the surface through the opening. The head comes in contact with the surface for read/write function. The head positing is of two types: first is the head cam positioning mechanism while the second is lead screw positioner. Both the methods use stepper motor for the drive. The hard disk spins at 4500 rpm and has an access time of 10 milliseconds. The tracks in the disk vary with its type. The 3.5 inch disks with a capacity of 1.44 MB can hold 80 tracks while a 3.5 inch disk of capacity 720 KB can accommodate 40 tracks.

Magnetic Tapes: Magnetic tape is a sequential address storage device. A typical magnetic tape is made of polyester coated with a magnetic material. The width of the tape is 0.5 inch and is stored in 10.5 inch reels. Each reel accommodates 2400 feet of tape. A tape transport system moves over the tape at a constant speed. When a particular read/write function is to be performed at a particular block, the tape moves at an increased speed and stops at the start of the block. But the high speed causes inertia; and due to this, the tape reel cannot be halted immediately. Therefore to avoid damage to the tape, whenever the tape is brought to halt it is pulled past the read/write head onto the capstan and idler wheel arrangement. Capstan is a polished metal piece. When the tape moves on the capstan, it is pressed by the idler wheel. This causes the tape to be pulled and it hangs in the loop between the two vacuum columns from each of the tape reel. The tape can be kept hanging by removing the air from the bottom of the column. To detect the presence and absence of tape loop, photoelectric sensors are located at the top and bottom of the columns. The length of the loop in the vacuum column is maintained with the help of these sensors which in turn control the reel motors. Therefore, when the tape moves faster on the reel, the loop grows downwards and reaches the sensor. The sensor then controls the speed of the motor thereby reducing the loop length. The cost of the tape drive can be reduced by avoiding the above mechanism. Such a tape drive is called streaming tape drive.

Magnetic Bubble Memory: Magnetic bubble memory is a serial access secondary memory storage device that was developed by Bell laboratories in 1968. Magnetic bubbles are cylindrical areas within specific magnetic material, gadolinium gallium garnet (GGG). It has uniaxial magnetic anisotropy and can be easily magnetized in a predefined plane. When an external magnetic field is applied to GGG along the specific plane then the areas that are not magnetized are shrunk. The shrinkage increases with the increase in the intensity of the applied magnetic field. The diameter of the bubbles is generally 10 micrometer and extends from 0.01 to 100 micrometers. The presence of a bubble is indicated as 1 and the absence is 0.

Optical Storage Systems: Optical storage technology is the current and future for large scale information storage. CD-ROM is a type of optical disk. They combine the optical storage and floppy disk technologies. They are available as read only and rewritable formats. An optical disk can store large amount of data on a removable disk. A 5.25 inch disk has a capacity of 2 GB. Unlike the Winchester disk or hard disk, optical disk is removable, robust, inexpensive and portable. They possess an archival life of 100 years.

The data is written on an optical disk with the help of a focused laser beam. The near infra red laser has small diffraction diameter and permits high recording capacity. The data can also be written using visible laser whose performance is much better. A highly focused laser beam is obtained from a laser diode. They beam is collimated by passing through a lens and passed onto the circulizer. The circularizer transforms the beam to cylindrical-shaped beam. The beam travels along a mirror that defects the beam by 45° and directs it to the objective lens. This lens focuses the beam into a limited spot on the optical disk. The recorder beam on the surface of the disk modifies the property of the lava beam. The change is deducted to read the binary data recorded on the disk. The structure of the optical disk is illustrated in Figure 15.4 is known as pre-grooved structure. The groove pattern is produced by the stamper. The stamper is complementary to the groove pattern on the disk. The manufacturing of stamper is done in the following three processes:

1. Photoresist process to form original master.
2. Electroforming process to make nickel plate with grooves and address data on the surface.
3. Finally, mechanical process to obtain the required structure.

Optical disks can be classified into different types based on the markings are recorded. CDs are 120 mm in diameter and are of three types: (1) CD-ROM, (2) CD-R

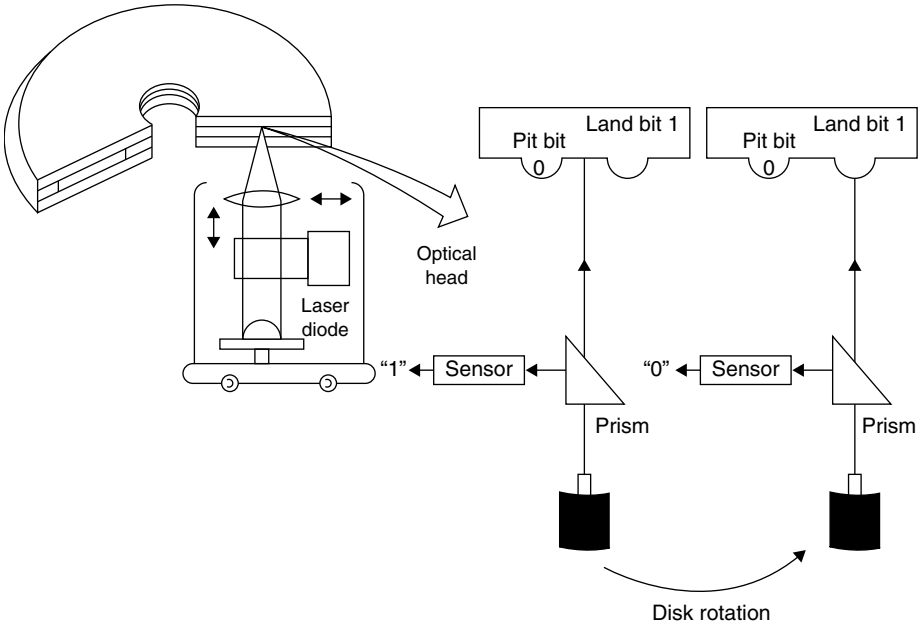


Fig.15.4 Structure of optical disk

(where R stands for recordable) and (3) CD-E (Erasable). A CD-ROM can store up to 700 MB of data and the speed is slow. The CD-R diskettes read and write information. The disk is used for storing large volume of information. The CD-E has the advantage of erasing and rewriting data. They have data capacity of 40 MB. It consumes less power and the data can be overwritten directly unlike the magnetic optical drives. Optical diskettes are basically magnetic floppy disk with an optical track that can simplify tracking with the help of laser beam. They store data up to 21 MB. They do not use laser or optical beam to read or write data and hence are not strictly optical.

INPUT–OUTPUT DEVICES

The devices that are connected to the computer externally are called peripheral devices. Peripheral devices are input and output devices and other peripherals. Input–output devices are hardware elements that enable in connecting the physical human effectors like hands, voice, etc., to the computers. These devices enable the communication between users and software. The purpose of an input device is to enter data into a computer system and for issuing instructions or command to a computer. The input devices transform data provided by the user into a suitable format that can be accessed by computer. The process is not performed by the input device but is combined with appropriate software called device drivers. A good input device should have swift time response and provide accurate useful data to the main memory of the computer for processing. Followings are the most useful input devices. The input and output devices of the computer are used for storing instructions, programs, etc., and for obtaining results. Secondary storage devices can be used for both input and output. Input devices are used to convert the input analog information to its binary form. The analog input is converted to digital form. Some of the important input devices are discussed as follows.

Keyboard

The keyboard is the simplest input device. It is similar to the typewriter used in the precomputer period. The working of a keyboard is simple. The keys in the keyboard detect the pressure of a finger and an encoder converts this into the corresponding binary code. The keys of a keyboard act as a switch that sets to ON state when they are operated. The structure of a mechanical key switch is shown in the Figure 15.5. When the key is operated, the pressure is transferred downwards, which in turn presses a spring. This closes the contact to an encoder circuit.

An audible click is generated by the steel snap disk that is placed between the plunger and contact wires. A similar sound is produced when the key is released. There are many types of switches like rubber dome switch, membrane switch, hall effect switch and capacitive switch. The signal from the keyboard is converted to unique ASCII code by the keyboard encoder circuit. The keyboard encoder consists of an input and an output port. These ports have a set of 8 terminals each. A 5V power supply is connected to each terminal of the port through a pull up resistor. Thus, a set of eight wires are obtained row wise and column wise thereby forming a matrix.

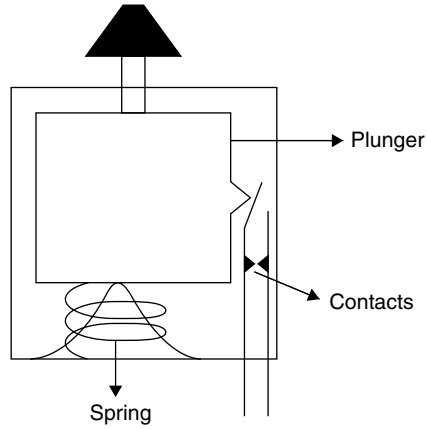


Fig.15.5 Key switch

Several kinds of keyboards are available, namely, QWERTY, Dvorak, alphabetical, chord and numeric keypad but the most common keyboard in use is the QWERTY board. The QWERTY keyboard uses the most common arrangement of alphanumerical keys and is essential for entering a variable data. Design of the keyboard is user friendly especially for right-handers and hence it is one of the most common used types. However, it may be time consuming for untrained users to master typing. The keyboard does not support the conventional 10-finger typing since the keys are arranged randomly. The second most commonly used keyboard is the alphabetical keyboard in which the keys are arranged in alphabetical order. Since the placement of the keys is in a predictable order, the beginners also find it much easier. But the alphabetical keyboard is much slower than the QWERTY or Dvorak keyboards and therefore avoided in PC applications.

Mouse

A mouse is an electro-mechanical, hand-held pointing device. They fall under the category of pointing devices that help in selection of entities on a computer display. They are used to navigate the cursor, identify the entity, select the desired entity and execute. Mouse is used in selecting menu commands, moving icons, resizing windows, starting programs, choosing options and many more. The structure of the mouse has a magnetically coated ball placed internally. The ball detects the movement of the mouse across the monitor. The mouse should be operated by placing on a flat smooth surface to avoid friction. The mouse is a small case that has three parts: the left navigator, right navigator and the central roller and generally can be held under the user's palm. The cursor moves very fast with the mouse and help to work in any direction. It is easier and faster to move through a mouse. The device gets its name due to its shape as it resembles a mouse. They are connected to the computer through a connecting wire that has a USB port at its end.

Types of Mouse

There are various types of mouse: mechanical, optical or cordless. Each of the type is described as follows:

Mechanical Mouse: Mechanical mouse has a ball in the centre that controls the movement of cursor on the computer monitor. The ball can be rolled in any direction and the movement is detected by a sensor and also moves the mouse pointer in the similar direction.

Optical Mouse: In an optical mouse, the movement of cursor on the computer monitor is controlled by laser rays. The movement of the cursor is detected by the reflected light. It uses a sensor that senses the changes in the reflected light rather than the motion of a rolling sphere.

Cordless Mouse: Cordless mouse does not possess a connecting wire for connecting with the motherboard. The cord less mouse is battery. The data is transmitted through infrared or radio signals. Mouse is essential for designing pictures, graphical data and video games by multimedia designers as it is much easier to navigate. Unlike the other mouse, an optical or laser mouse does not require a mouse pad.

Mouse has replaced the much complex key combinations that were used for "point and click" actions. Nevertheless, it cannot alternate all keyboard operations and is restricted to the command-based operations.

Trackball: Trackball is similar to a mouse in all its functions and is used as a pointing device. The trackball is stationary and one does not need to navigate the entire device on a mouse pad. It has a rotatable ball at the top of the stationary device that can be moved to navigate the pointer. In a trackball, the ball placed on the top can be rolled with the fingers. In addition to the movable ball, the trackball has two or three buttons used to select items from the screen.

Joystick

Joystick is a stick-shaped remote control device connected to a platform that is movable in all directions. They are mainly used for playing video games, cranes, trucks, underwater unmanned vehicles, flight simulators, industrial robots, etc. The motion of the joystick corresponds to the movement of objects or characters in computer games. Usually joysticks have buttons also called triggers. The joystick has a ball at the end that moves in a socket. The user moves the stick by holding the ball to control machines.

Touch Screens

Touch screens are monitors or electronic visual display screens that are used on portable computers and laptops. The screen senses the finger touch and moves the pointer accordingly.

The touch screens have the same function as the mouse buttons. The touchpad is a great option for persons with disability, who have limitations in moving their hand. The desired selection can be chosen by directly touching the screen instead of moving

a cursor to the point on the screen as in the case of a mouse or joystick. Touch screens have replaced the commercial usage of mouse, for example, ATM machines, railway enquiry, cell phone screen, etc.

Light Pen

Light pen is a light-sensitive input device that uses a light detector to select objects on a display panel. It is connected to the computer terminal through a wire. It detects the CRT beam when the pointer touches the screen due to the phosphors present in the screen. This generates a narrow electrical pulse which is fed as the input signal to the computer. The function of a light pen is similar to a mouse but the difference being that a light pen can move the pointer to select objects on the display screen by directly pointing to the objects on the screen. A light pen can work with any CRT monitor but not with LCD monitors. It is used by architects and engineers for CAD applications and editing.

Barcode Reader

Barcodes are small labels containing white and black stripes of varying width on consumer products. These labels contain information on the product. When the barcode is placed in front of a special reader such as a light pen, it detects the details and displays it on the computer monitor. The barcode reader consists of a light emitting diode that will emit light on the code. The reflected light from the barcode is transmitted to the computer by a light-sensitive diode.

Optical Code Reader (OCR)

Optical mark reader reads data in specific places on specially prepared cards. They are used in cards that are shaded by pencil. These are later detected by the detector that enables to read the information.

Magnetic Ink Character Reader (MICR)

Magnetic ink character reader is a device used to write numbers with coded magnetic ink. This special ink contains the written numbers magnetically. These numbers written are fixed and cannot be changed. MICR is much faster than OCR or other methods when reading codes. This system is widely used on cheque books issued by banks.

OUTPUT DEVICES

The input data after processing enters the next phase called the output device. Output devices deliver the processed information back to the user. Some of the commonly used output devices are:

- Monitor
- Printers
- Plotter
- Speech input unit

Monitor

Monitor is one of the most important output devices in the computer system. The computer monitor displays programs, active applications and data stored in the computer system. Monitor is the visual display unit of the computer system and displays video images. The images displayed are not permanent and is closed on stopping a program. The display devices store data in the form of soft copy. The video adapter card and monitor is used to display the computer information visually. The processed data from the CPU is sent to the video adapter that converts it to visual data and is displayed. Two basic types of monitors used with microcomputers are CRT and LCD.

Cathode Ray Tube (CRT): Cathode ray tube monitor is the typical monitor that is seen on a desktop computer. The appearance and working of CRT is similar to a television screen. It uses a large vacuum tube called cathode ray tube (CRT).

Liquid Crystal Displays (LCD): The flat display panel makes use of the liquid crystal displays. It comprises of liquid crystal that helps in rendering images. Owing to its resolution and picture clarity, they have gained popularity in recent times.

The resolution of the monitors is highly important and generally monitors have a resolution of at least 800×600 pixels. High-end monitors can have resolutions of 1024×768 pixels or even 1280×1024 pixels. A detailed description on the different type of display devices is dealt with in Chapter 13 of this book.

Printer

The output is given on paper called hard copy and these are produced by printers. Printers can be classified into impact or non-impact printers.

Impact printers use the typewriting printing mechanism wherein a hammer strikes the paper through a ribbon to produce output. Dot matrix and character printers fall under this category.

Non-impact printers do not touch the paper while printing. They use chemical, heat or electrical signals to etch symbols on paper. Inkjet, laser, thermal printers fall under this category of printers.

The resolution and speed are two main characters about printers. Print resolution is measured in terms of number of dots per inch (dpi). Print speed is measured in terms of number of characters printed in a unit of time and is represented as characters-per-second (cps), lines-per-minute (lpm) or pages-per-minute (ppm).

Some of the most commonly used printers are laser printer, ink jet printer, dot matrix printer and line printer.

Dot Matrix Printers: Dot matrix printing is a primitive type of printing technique. It uses heat sensitive paper to print the data. The print head "hits" the paper across the ribbon that is placed between the needle and paper. The hitting of the print head produces text (or images) by combinations of dots; hence the name dot matrix printer. The printing process is slow and produces screeching noise during printing. It is a conventional and inexpensive output device. Although it is inexpensive, it cannot print many papers simultaneously and produces lower print quality and it is outdated.

Inkjet Printers: These printers have a sprayer that sprays the ink in the form of aerosol through tiny nozzles, which gets charged when passed through an electric field. The paper absorbs the ink and it dries it instantaneously. These printers are expensive but faster and can support colour printing by using ink in various colours. The image quality is not sharp when compared to the output of a laser printer yet the quality of inkjet images is still high. In general, inkjet printer offers an excellent middle ground between dot matrix and laser printer. The inkjet printer produces less noise while printing yet it is slow compared to the laser printer. Typically, an inkjet printer is more expensive than a dot matrix printer, but costs only half as much as a laser printer.

Laser Printers: The printer has a drum containing black toner powder that is positively charged. When the laser beam is passed across the drum it gets negatively charged and attracts the toner powder which in turn sticks to the drum. When the paper moves on the drum, the toner powder is transferred to the paper. The laser printer is extremely fast and quiet. The printing speed is about 15,000–20,000 lines per minute. The intake of paper is automatic. It produces high quality print without smudging of ink.

Electrographic Printer: This printer writes on special paper using electricity. The paper is held between two electrodes. These printers can either use wet process or dry process. The speed of these printers is fairly high , that is, about 20,000 cps. These printers are very reliable as there are very less moving parts.

Electro Static Printer: This printer utilizes static electricity to print information on paper. The toner powder is attracted to charged spots thereby printing the information. These printers are very fast and are capable for plotting graphs. The speed is up to 20,000 lines per minute.

Plotter

A plotter is a unique sort of output device that generates images on paper. These are designed so as to produce large drawings or images, construction plans for buildings or blueprints for mechanical objects. A plotter has a similar port as the printer. It comprises of an arrangement of coloured pens in a clip rack and a movable robotic arm. The instructions such as the desired colour, start and finish point coordinates for a line are fed to the computer. The plotter once on receiving the information selects the mentioned coloured pen with the help of the robotic arm and positions it according to the given coordinates. The pen is lowered to the surface of the paper and draws the line until it reaches end coordinates. Plotters draw curves by creating a sequence of very short straight lines.

Speech Input Unit

The speech input unit converts the input in the form of sound to a data form that is easily understood by the computer. The data is converted into a unique string of bits that can be interpreted by the computer. The speech inputs can be single word by word

or continuously. It is essential that the computer recognizes both forms. The speech input may be further classified into speaker dependent and speaker independent. In case of speaker dependent, the computer recognizes the speech of a particular user and the spoken input by any user is identified by the speaker-independent systems. The speaker-dependent units are easier and are designed by using limited vocabulary since it recognizes only one user. Whereas the speaker-independent unit is more extensive as it should able to identify the speech input of any user. Thus, the designing of such a unit is more complex.

COMPUTER SOFTWARE

Computers rely on users for instructions for every task. These instructions are given to the computers in a specific manner or sequence that can be easily understood by the computers. These sets of instructions are called programs. A collective set of programs is known as software. Software enables the hardware to perform a particular set of tasks in particular sequence. Software can be classified mainly into the following categories and sub-categories as shown in Figure 15.6.

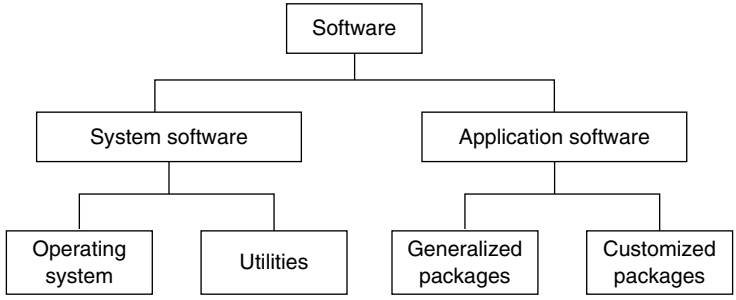


Fig.15.6 Classification of software

System Software

Computers when switched on initiates the programs stored in ROM that are executed to activate different units of it to make it ready for the user to work on it. System software is a collection of programs that are responsible for the functioning and controlling of the computer. Operating system (OS) falls under this category. The OS acts as an interface between the user and the computer. Hardware devices such as disk drives, keyboard, monitor, etc. are administered by the OS. It facilitates in maintaining the disk file systems and supports application programs. UNIX, Windows and Linux are some of the examples of OS. Although the OS governs the operations of the computer, they lack to fulfill the user's expectation. The shortcomings faced by system software led to another type of system software called utilities software. Utility programs are a broad category of software such as compress (zip)/uncompress (unzip) files software, antivirus software, etc.

Application Software

Application software is a set of programs that perform specific tasks. This software is created by the programmer to meet the specific needs of a user. These can be commercial and non-commercial programs. For example, An application package for managing library records, patients record, etc. Application software can be broadly classified into two types:

1. Generalized packages
2. Customized packages

Generalized Packages: These softwares are written to cater to user's general needs such as preparing documents, drawing pictures, database to manage data/information, preparing presentations, listen songs, watch movies, play games, etc. These programs are a collection of tools that provide solutions to solve specific problems. Some of the generalized packages are listed as follows:

- **Word processing software(for preparing documents):** MS-Word, openOffice.org Writer.
- **Spreadsheets (data analysis):** Lotus Smart suites, MSExcel, OpenOffice.org Calc, Apple Numbers.
- **Presentations:** Presentation Graphics, MS-PowerPoint, OpenOffice.org Impress.
- **Database management system:** MS-Access, OpenOffice.org Base, MS-SQL Server, ORACLE.
- **Graphics tools:** Paint, Adobe Photoshop.

Customized Softwares: Customized softwares are developed to meet the specific requirements of an organization/institution. They can be written and necessary modifications can be made by the programmer. These packages are developed using high-level computer. For example, Payroll packages, inventory control, etc.

COMPUTER LANGUAGES

Computer languages are means of communication between the user and computers. The computer language is easily comprehended both by user and the machine. Every computer language is bound by a set of rules known as SYNTAX of the language. The user is restricted by that syntax while communicating with the computer system. Program is a series of instructions that enable the computer to work.

Computer languages are broadly classified into low level language and high level language.

Low Level Language

The term "low level" means simple instructions in the language that can be easily understood by the machine. Low level languages are further classified into machine language and assembly language.

Machine Language

Machine language is the most uncomplicated level of programming language. Machine language is interpreted and executed directly by the computer. Every computer has its own machine language and relates only to its machine language. These are generally provided by the manufacturer. This language is coded in the form of string of 0's and 1's that form the basis of operating computers. For example, 10010101. the absence of electric pulse is denoted as "0" and the presence of an electric pulse is denoted as "1". The computer recognizes the electric signal and hence it is dependent on machine language. It is difficult to write programs since the binary codes are not user friendly. The user is forced to keep track of the locations where the data and instructions are stored.

Assembly Language

Assembly language is also mentioned by the manufacturer. Every computer has individual set of instructions in assembly language. Unlike the machine language the instructions in assembly language are provided as a set of letters instead of binary digits. These set of letters that represent an instruction is called mnemonics. Since the program is assembled into the machine for execution this is known as assembly language. The formulation of the assembly language brought in improvizations in programming. Assembly language programming is simpler and less time consuming than machine level programming. Troubleshooting is much easier in assembly language than in machine language programs. Low level language occupies less storage space. It is extremely useful in writing programs involved in operating system since they must run fast. Although the assembly language programming is less complex, the disadvantage of this programming method is that it is also machine dependent, hard to learn, time consuming and programs are long. Also programmers must have prior knowledge on the compatibility of the machine on which the program will run.

High Level Language

Low level language requires widespread understanding of the hardware since it is machine dependent. To overcome the limitation of low level language, high level language was evolved. The advent of the high level programming which uses normal English, made easier programming and the user was able to understand instructions and rectify any problem. Higher level languages are computer independent and are problem oriented. Since they are written in English, the high level language is simple to write, read and execute. Modification in the program is much easier. A single statement is applicable for several computer operations and hence the program is short and its execution faster. They can be executed by any computer as they are program independent. The various high level languages are given as follows:

FORTRAN: FOTRAN was developed in 1956 with the objective of solving mathematical and scientific problems. It is derived from FORmula TRANslation. It is one of the most popular languages among scientific world and has the following characters:

FOTRAN Characters:

- **Alphabetic:** A B C D E F G H I J K L M N O P Q R S T U V W X Y Z (26)
- **Numeric:** 0 to 9 (10)
- **Special characters:** (11)
  "+" plus,
  "–" minus,
  "/" division or slash,
  "*" asterisk,
  "=" equal to,
  "." decimal,
  "," comma
  "(" open parenthesis
  ")" close parenthesis
  ‘’quote mark
  $ dollar symbol

FORTRAN has many versions FORTRAN II, FORTRAN IV, FORTRAN 77 being the latest.

COBOL: COBOL is one of the oldest high-level programming languages and its name is derived from Common Business Oriented Language. It was developed in 1959 and has gone through numerous modifications and improvements. COBOL: It is a standardized language used for commercial applications. Despite its shortcomings, it is still considered to be the most popular commercial language. The COBOL characters are mentioned below. Since 1959 it has undergone several modifications and improvements. The most standard version is ANS COBOL.

COBOL Characters:

- **Alphabetic:** A B C D E F G H I J K L M N O P Q R S T U V W X Y Z (26)
- **Numeric:** 0 to 9 (10)
- **Special characters:** (15)
  "+" plus,
  "–" minus,
  "/" division or slash,
  "*" asterisk,
  "=" equal to,
  "." decimal,
  "," comma
  "(" open parenthesis
  ")" close parenthesis
  ‘’quote mark
  $ dollar symbol
  ">" greater than
  "<" lesser than

":" semicolon
" " Blank

BASIC: The name is derived from Beginners All purpose Symbolic Instruction Code and was invented in 1963 by John George. It was used as a teaching tool for students. Owing to its simple programming style, it is the most commonly and widely used computer programming languages. It is highly suitable for beginners compared to the powerful languages such as FOTRAN. BASIC gained its popularity with microcomputers in earlier days. BASIC is a multi-platform language and allows loops in the programs. It permits subroutines, built-in functions, user defined functions, arrays, sorting and searches. This program language can be used for interactive and hands-on practical purpose. They can be used both by professionals and non-professionals.

BASIC Characters:

- **Alphabetic:** A B C D E F G H I J K L M N O P Q R S T U V W X Y Z (26)
- **Numeric:** 0 to 9 (10)
- **Special characters:** (20)

  "+" plus
  "–" minus
  "/" division or slash
  "*" asterisk
  "=" equal to
  "." decimal
  "," comma
  "(" open parenthesis
  ")" close parenthesis
  ‘’quote mark
  $ dollar symbol
  ">"greater than
  "<" lesser than
  ":" semicolon
  : colon
  % percentage
  ? question mark
  # hash symbol
  ^ exponential operator
  " " Blank

ALGOL: ALGOL, the short form for ALGOrithmic Language, is one of the vital computer programming languages developed in the 1950s. It was used for standard representation of algorithms. It was the most successful high level programming language compared to FORTRAN and COBOL. The problems in using FORTRAN were overcome by ALGOL which later introduced PASCAL and C programming. ALGOL has three major specifications, namely, ALGOL 58, ALGOL 60 and ALGOL 68. The latest

version is ALGOL 68 that has the flexibility in arrays and introduced extensive features that were used in the programs.

Pascal: Pascal programming language was named after the mathematician Blaise Pascal. It was originally developed by Niklaus Wirth with features that were lacking in other languages. Pascal was developed to improve the efficiency to implement and run and develop well-structured programs that can be used for explaining essential concepts of computer programming. Pascal programming was incorporated with increased features like built in type data in the form of integers, real character and Boolean type. Pascal programming language is known for its simplicity and expressivity. It serves as a starter level for C programming.

C Language: C programming is a general purpose high level language that was formalized by Dennis Ritchie in 1972. It was developed mainly to write an operating system called UNIX. C programming is one of the most widely used professional languages for various reasons. The programming is structured and is easy to comprehend. The programs are highly efficient and can be used even for low-level activities. They have high compatibility and can be compiled on a variety of computers. C is one of the most widely used system programming language. C is a powerful and flexible language that provides fast program execution and minimizes restrictions on the user.

Translating Computer Languages : Compiler and Assembler

Assembly language is machine independent and uses mnemonics to convey instructions to the computers. These mnemonics cannot be comprehended by the computers and requires translators. Compilers and interpreters convert the instructions for later execution by real hardware or software interpreters.

Compilers: Compiler is a software or set of codes that are used in converting a high level language such as C, Java, etc., into the binary form that can be easily understood by computer hardware. These softwares are used for interpreting high level languages and executing them to perform various tasks. A specific high level language has a specific compiler/interpreter designed. When a programmer writes a code in a high level language like C and desires to execute it, then a specific compiler specially designed for C is used for execution. The compiler primarily scans the entire program and later translates it into machine code. The resultant machine language is executed by the computer processor and the corresponding tasks will be performed. The program in the high level language is called source program and the program generated by the compiler after translation is called object program. Although the compiler and interpreters differ by their names, they have the same objective and are intended in converting high level languages into machine language. Assembler is a translation software that converts assembly language to machine code. It is a type of compiler.

Interpreters: Interpreters are softwares whose functions are similar to compilers. These are also used to convert high level language into machine comprehensible binary

equivalent. However, the process of conversion is slightly different from compilers. When an interpreter receives a high level language code to be translated, it first converts it into an intermediate code before converting it into a machine code. Each part of the code is interpreted and executed discretely in a particular sequence. If an error occurs during the execution, then it will stop the interpretation of the code without translating the next set of the codes.

Table 15.1 Differences between compiler and interpreter

Compiler	Interpreter
The entire program is considered for translation using a compiler.	The input program is translated line by line using an interpreter.
The input high level language program is directly converted to machine language.	An intermediate object code is generated before converting to machine language.
Control instructions are executed faster.	Control instructions are executed slower.
Memory requirement is low.	Memory requirement is high.
Errors are displayed after entire program is checked.	During translation, errors are displayed for every instruction interpreted.

MULTIPLE CHOICE QUESTIONS

1. Which of the following is used as monitors for computer screens?
   (a) Current ray tube (b) Current ray technology
   (c) Cathode ray tube (d) Cathode ray technology
2. Which of the following secondary storage device is mounted on "reels"?
   (a) Floppy disk (b) Hard disk
   (c) Magnetic tapes (d) CDROM
3. A typical computer operation uses _____ as it memory device.
   (a) RAM (b) ROM
   (c) LED (d) CRT
4. A program to be translated into machine language is called _____.
   (a) Object program (b) Source program
   (c) Mnemonics (d) None of the above
5. The ALU, CU and MU together form the _____ of the computer.
   (a) Memory device (b) CRT
   (c) CPU (d) None of the above
6. Which component of the computer is responsible for interpreting and executing program instructions?
   (a) Storage unit (b) Logic unit
   (c) Control unit (d) None of the above

7. Codes used in assembly language are known as _____.
   (a) Codes (b) Mnemonics
   (c) Assemblers (d) Compilers
8. The computer easily comprehends _____.
   (a) High level language (b) System program
   (c) Assembly language (d) Machine language
9. An optical input device that interprets pencil marks on paper media is
   (a) O.M.R. (b) Punch card reader
   (c) Optical scanners (d) Magnetic tape
10. COBOL is a _____.
   (a) Low level language (b) High level language
   (c) Assembly language (d) Machine language

DESRIPTIVE QUESTIONS

1. Discuss the evolution of computers with a special note on microprocessors.
2. Describe the classification of computers based on generations.
3. What is a computer? Draw a block diagram to illustrate the basic organization of a computer system and explain the functions of various units.
4. Classify computers based on hardware, usage and size.
5. Discuss the various input and outputs devices of a computer.
6. Discuss in detail the various types of secondary storage devices.
7. Discuss in detail the types of monitors: CRT and LCD.
8. What is a software? Discuss its various categories with the help of a neat diagram.
9. Distinguish between systems and application softwares.
10. Write short notes on
   (a) Laser printer (b) High level language
   (c) Floppy disks (d) High level languages
   (e) Compiler (f) Mouse
   (g) Trackball (h) Plotter
   (i) Optical storage devices
11. Discuss in detail computer language and its classification.
12. Distinguish between compiler and interpreter.
13. Discuss in detail the various applications of computers.

ELECTRONIC EXERCISES

CMOS GATE – CD 4001 NOR

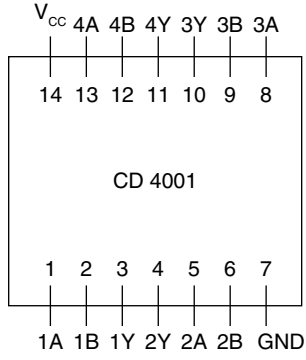
Aim

- To prove that NOR gate is a universal gate using CMOS NOR
- To construct gating circuits using CMOS NOR
- To construct an stable multivibrator using CMOS NOR

Procedure

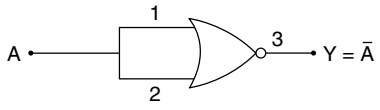
NOR Gate – Universal Gate: To prove that NOR gate is a universal gate using CMOS NOR. We connect or make use of the IC 4001 as a NOT by shorting the 2 inputs and obtained the output $\overline{A}$. In this the pin 14 is connected to +ve and the 7th pin is grounded. We also ground 5, 6, 8, 9, 12, and 13.

Pin Configuration:



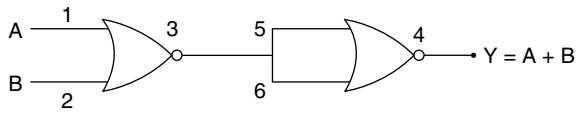
NOR Gate – Universal Gate:

NOT



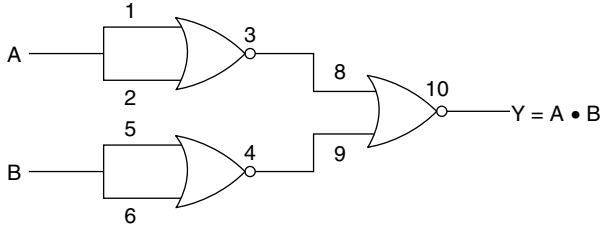
A	Y = Ā
0	1
1	0

OR



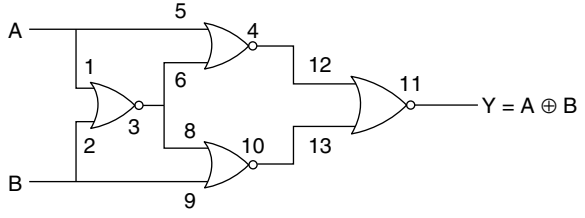
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

AND



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

XOR

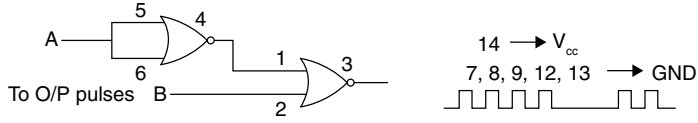


A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

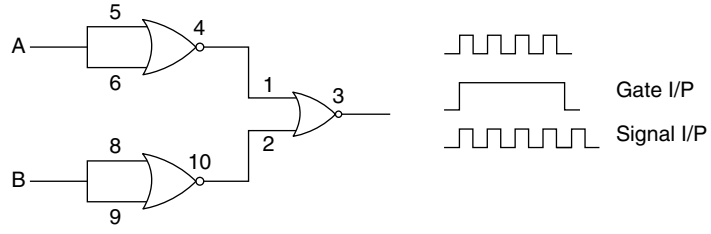
The NOR chip is then connected to test OR function by connecting a NOR gate connected to a NOT gate and checking the truth table. Similarly, it is checked for AND function by using 2NOR gates as NOT gate. Similarly, the NOR gate is also designed to perform Ex-NOR function and prove that it is a universal gate.

Pulse Disabling and Enabling Gate Using CMOS NOR: The circuit for pulse disabling gate was a NOR gate as a NOT to which autoclock is connected. The output is given as one of the input to a NOR gate. It's another input is connected to the pulser. The output from this is considered as a required output. Similarly, the circuit for the pulse disabling gate using CMOS NOR is constructed and its output is checked.

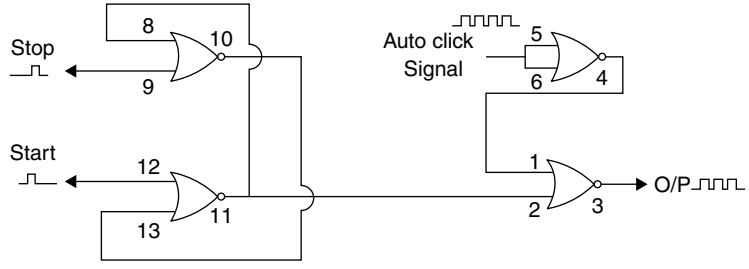
Pulse Disabling Gate



Pulse Enabling Gate



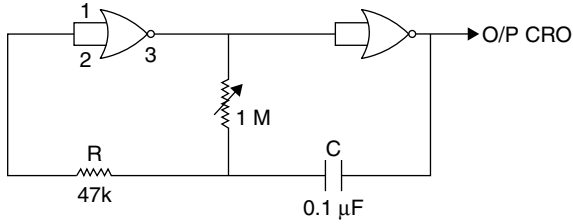
Electrically Triggered START/STOP Gate: The circuit is shown in the diagram. The circuit connected to the V_{CC} is connected to pin 14 and the ground to pin 7. The output which is electronically triggered pulse is obtained at pin 2.



Astable Multivibrator: The circuit is connected as shown in the diagram. A resistance of 1MSL and a capacitor of 0.1μf is connected. Its experimental frequency is obtained using a CRO and their frequency is obtained using the formula

$$f = \frac{0.45}{RC}$$

where R and C are resistance of capacitance used.



Result

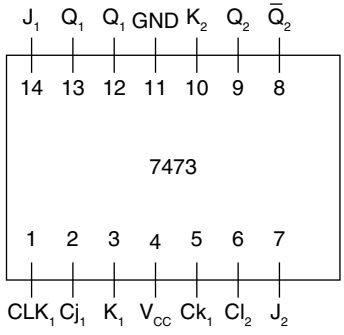
NOR gate is proved to be a universal gate using CMOS, NOR and gating circuit are constructed and A stable multivibrator was also constructed.

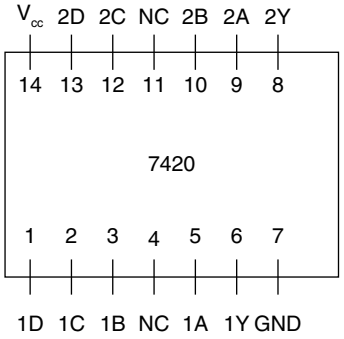
MODULUS COUNTER – JK FLIP-FLOPS

Aim

To construct

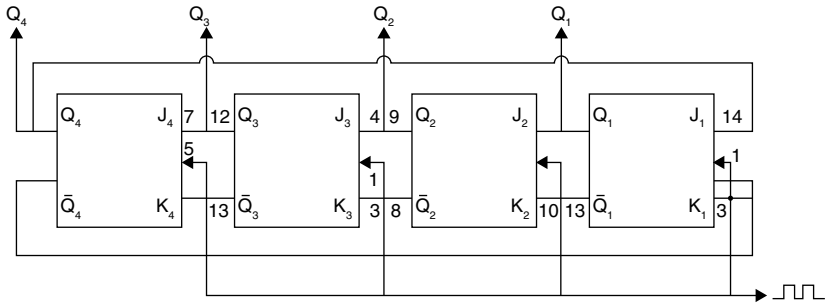
- Ring counter.
- Shift counter (odd and even modulus) using shift register configuration.
- An asynchronous modulus counter using JK flip-flops employing feedback technique.





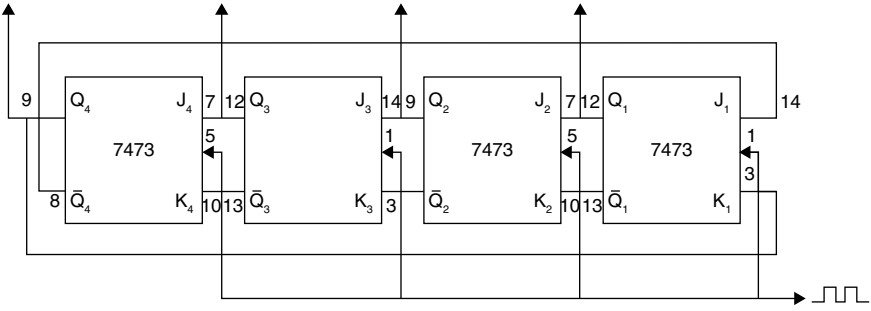
Procedure

MOD-4 Ring Counter: The circuit involves two IC 7473 chips. The clock is connected to first and 5th pin of the 2nd IC and the clear terminals 2nd and 6th pins of the ICS are connected to +5volts. As a feedback technique, the 14th pin of the first clip is connected to $\overline{Q}_4$. When the pulse are given, the result truth table is verified.



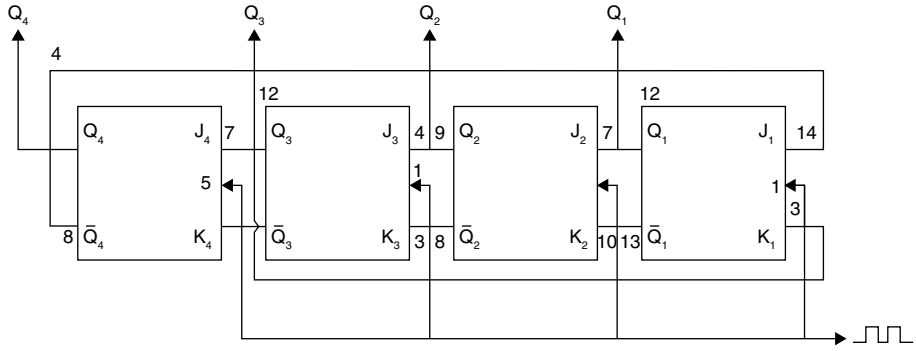
Q_4	Q_3	Q_2	Q_1
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	0

Even Modulus Shift Counter: The circuit is shown in the diagram. The connections are made. The connections are similar to the connections to MOD-4 ring counter except that k_1 is connected to Q_4 instead of $\overline{Q}_4$ and J_1 is connected to $\overline{Q}_4$ unlike the above case.



Q_4	Q_3	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1
1	1	1	0
1	1	0	0
1	1	0	0
0	0	0	0
0	0	0	1

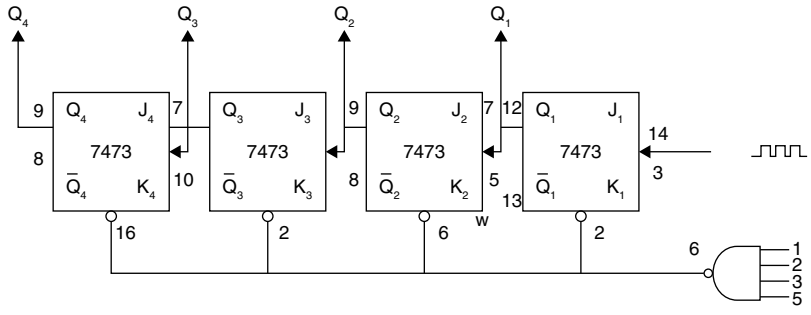
Odd Modulus Shift Counter: The circuit is connected as shown. In this circuit, J_1 is connected to $\overline{Q}_4$ similar to proceeding case but K_1 is connected to the clock of the J_1, K_4 flip-flop. This circuit acts as odd modulus counter, that is, for modulus 3, 1, and 5.



Q_4	Q_3	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0
0	0	0	0

Asynchronous Counter: The connections are made as in the circuit. Were the clock is given to the pin 7 of IC7473. This clock drives the circuit and the outputs are connected from their other clock. The clear terminals are connected to the (pin 6) of the NAND gate and the input from Q_1, Q_2, Q_3, respectively are given to 3rd, 4th, 5th pins of NAND gate.

JK Flip–Flops Employing Feedback Technique:



Result

Asynchronous counter using JK flip-flops employing feedback technique was constructed. Ring counter and shift counter were also constructed.

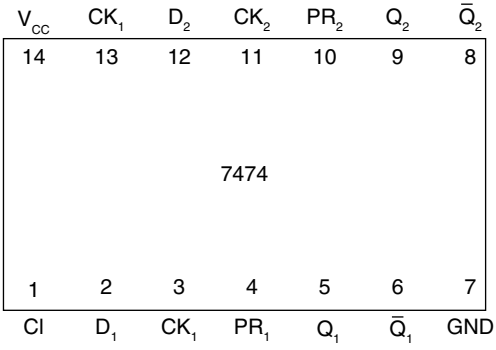
MODULUS COUNTER – D FLIP-FLOPS 7474

Aim

Using D flip-flops construct

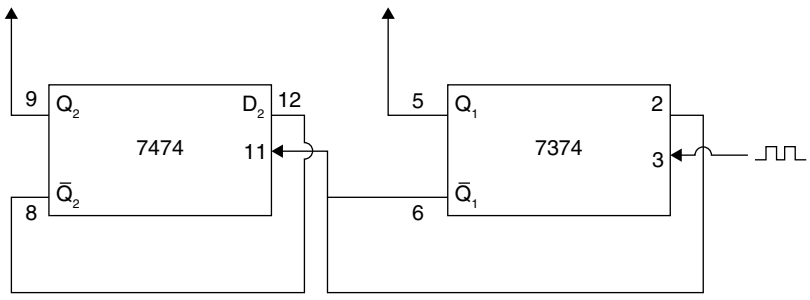
- MOD-4 UP, DOWN counters using feedback technique
- MOD-3 ring counter using shift register configuration.

Pin Configuration



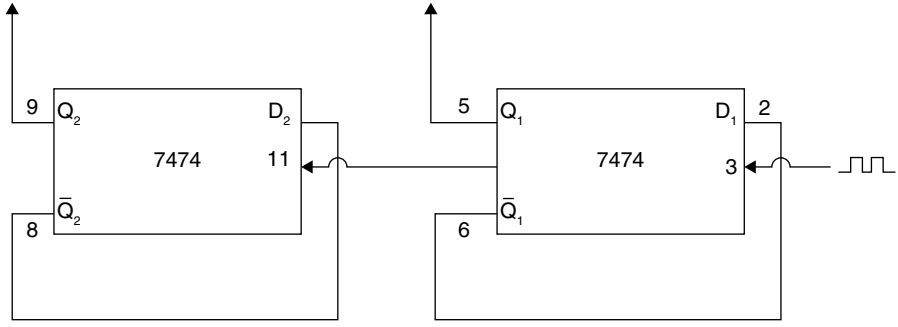
Procedure

MOD-4 UP Counters: The connections are wired up as shown in the circuit diagram. A D flip-flop which is 7474 IC is used. The clock is given to pin 3 of the D flip-flop and the output from D_1 (i.e.) pin 2 is given or it drives the clock of D_2. It is also connected to $\bar{Q}_1$. Similarly, the output from D_2 is Q_2. As the clock pulses are given, the truth table is verified.



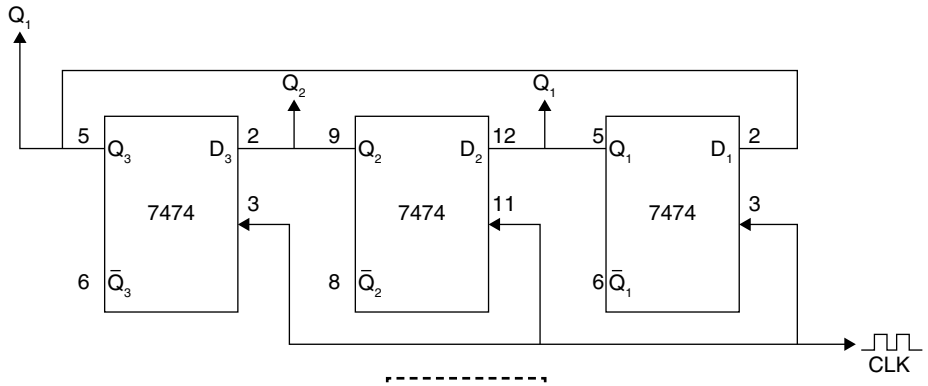
No. of pulses	Q_n	
	Q_2	Q_1
0	0	0
1	0	1
2	1	0
3	1	1

MOD-4 DOWN Counter: The connections are wired up as shown in the diagram. Here the clock is given to D_1 and the output from D_1 is connected to $\overline{Q}_1$. Here unlike the above case, Q_1 drives the clock of D_2, instead of $\overline{Q}_1$. Similarly, the output from D_2 is given as input or connected to $\overline{Q}_2$ and the outputs are obtained from Q_1 and Q_2. By connecting this way, the counter starts downward counting on giving the clock pulse the result or the outputs are verified with the truth table.



No. of pulses	Q_n	
	Q_2	Q_1
0	0	0
1	0	1
2	1	0
4	0	0

MOD-3 Ring Counter: This set up makes use of two IC 7474 chips. The connections are wired up as shown in the circuit diagram. In this circuit, the output from one is given as the input to the other (i.e.) the output from Q_1 is given to D_2 and that of Q_2 is given to D_3. On giving the pulses of the output is verified. The circuit thus acts as a MOD-3 ring counter.



Q_n		
Q_3	Q_2	Q_1
0	0	1
0	1	0

Result

The MOD-4 UP and DOWN counters using D flip-flop MOD-3 ring counters were constructed and verified.

MODULUS COUNTERS – JK FLIP-FLOPS – SYNCHRONOUS

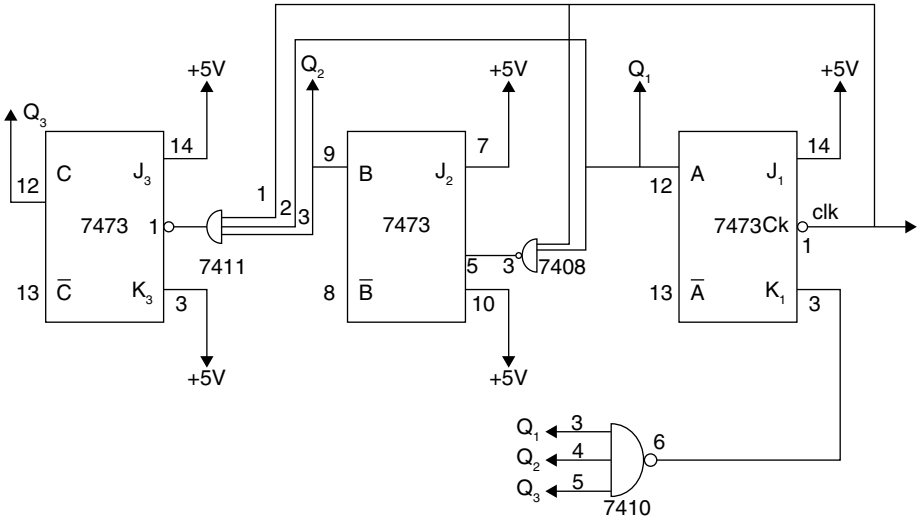
Aim

To construct synchronous modulus counters MOD-7, MOD-6, MOD-5 using JK flip-flops and required logic gates.

Procedure

The experiment involves the construction of a MOD-7, MOD-6, MOD-7 counters. The circuit consists of a pair of 7473 JK flip-flops. J_1 of the 1st flip-flop is given to +5V. The clock is given to pin 1 and pin 5 of the first flip-flop. The output from the 1st Q_n is given as one of the inputs to the 3-input AND gate whose other input will be driven clock. The output from the AND gate is given as the input to the clock pin 5 of the 7473 of the JK. The same connection type is given to the clock of the 2nd flip-flop. The outputs Q_1, Q_2, Q_3 are connected to output pins. The circuit also consists of 3-input NAND gate. The 3 outputs Q_1, Q_2, Q_3 are connected to the output pins. The output from the NAND gate is given to the K terminal of 1st JK flip-flop. Here the terminals 2, 6, and 2 of the first and second flip-flop, respectively.

MOD-7 Synchronous Counter: For MOD-7 synchronous counter, the output from the NAND gate should be connected to K_1. Hence, we skip 000 state. As the pulses are given, the modulus counter starts counting till the 7th state and the state 000 is skipped for MOD-7 synchronous counter.



No. of pulses	Q_3	Q_2	Q_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

MOD-6 Synchronous Counter: The output of K is connected to K_2. Here, 2 states are skipped i.e., 000 and 001. As the pulses are given, the synchronous counters start counting by skipping the above mentioned states.

MOD-5 Synchronous Counter: The output of the NAND gate is connected to K_1 and K_2. Here 3 states are skipped i.e., 000, 001, and 010. As the pulses are given, the counter starts counting.

Result

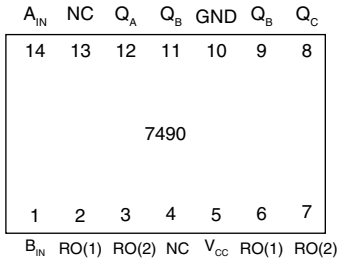
The synchronous modulus counters have been constructed as MOD-7, MOD-6, MOD-5 using JK flip-flops and logic gates.

MODULUS COUNTER – DECADE COUNTER (7490)

Aim

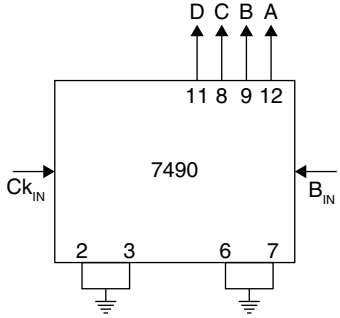
- To study the different features of the decade counter IC 7490
- To construct different modulus counters using IC 7490.

Pin Configuration:



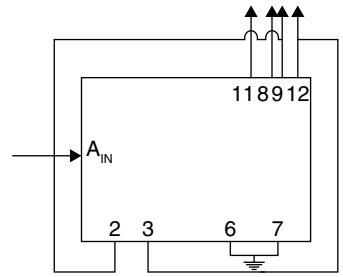
Procedure

MOD-3: To obtain a MOD-3 counter the clock is connected to pin 14 and the output is obtained from 12. There are 2 reset pin Ro(1) and Ro(2) which are connected to pins 2 and 3; similarly, there are other 2 reset pins Rq(1) and Rq(2) are connected to 9 and 12, respectively, through BD_{in}.



Q_2	Q_1
0	1
1	0

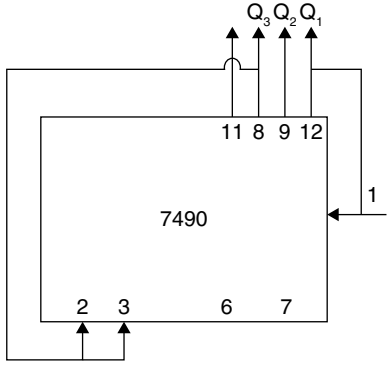
MOD-2 counter



Q_2	Q_1
0	0
0	1
1	0

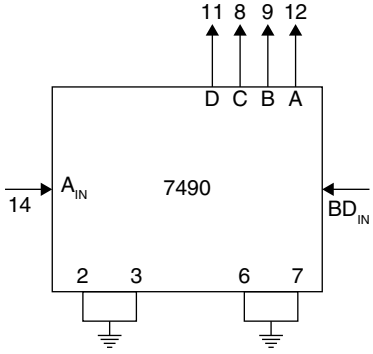
MOD-3 counter

MOD-4: To act as MOD-4 the IC 7490 is connected in the manner. Here Ro(1) and Ro(2) pins are shorted and is connected to pin 8 and the pin 12 is connected to BD_{in}.



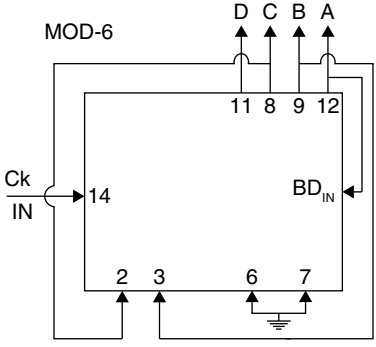
C	A	A
Q_3	Q_2	Q_1
0	0	0
0	0	1
0	1	0
0	1	1
0	0	0

MOD-5: In case of MOD-5 counter, the reset pin Ro(1) and Ro(2) are connected to 11 and 4 with 12 connected to BD_{in}.



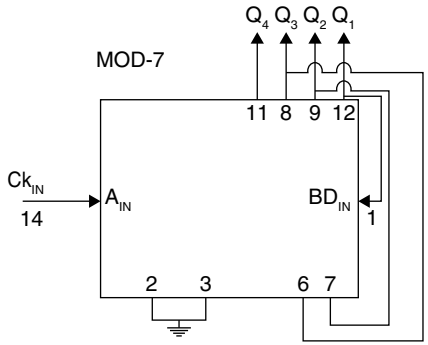
Clock	Q_3	Q_2	Q_1
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0

MOD-6: The MOD-6 is constructed as shown in the circuit. Pin 2 is connected to 8, 3 is connected to pin 0, 6 and 7 are connected together. Clock is connected to pin 14.



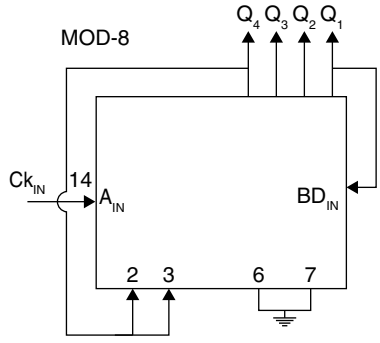
Q_4	Q_3	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
0	0	0	0

MOD-7, MOD-8, MOD-9: To use the decade counter IC 7490 as MOD-7 the reset pin Ro(1) and Ro(2) are get and the reset pins Ro(1) and Ro(2) are grounded.

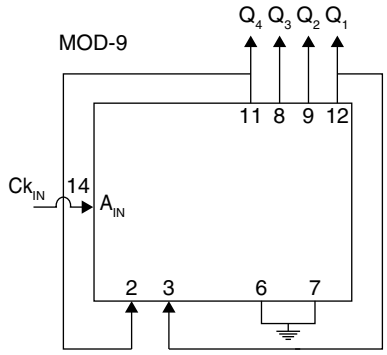


Q_4	Q_3	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
1	0	0	1
0	0	0	0

The IC 7490 can be made to act as MOD-9 counter by connecting the reset Ro(1) and Ro(2) to 11 and 12 respectively. The pins 11 and 1 are shorted.



Q_4	Q_3	Q_2	Q_1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
0	0	0	0



Q_3	Q_2	Q_1
0	0	0
0	0	1
0	1	0
1	0	0
1	0	1
0	0	0

Result

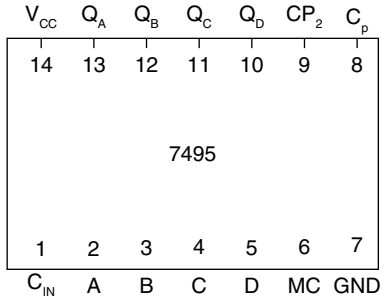
The different features of the decade counter IC 7490 are studied and different modulus counters are also constructed.

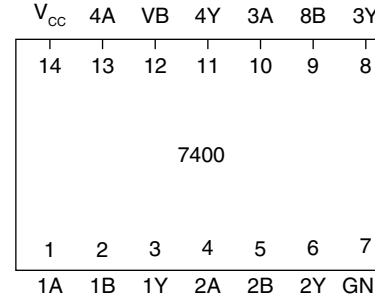
STUDY OF 8-BIT SHIFT REGISTER (7495)

Aim

To study the shift left, shift right capacitances of a 4-bit shift register IC 7495 and construct a MOD-4 ring counter and MOD-8 counter.

Pin Configuration:

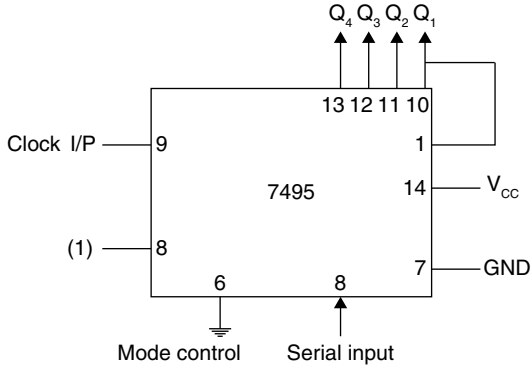




Procedure

Right Shift Operation: To obtain the right shift operation, the 14th pin is connected to V_{CC} and 7 is grounded. The pins made of control outputs are connected from Q_A, Q_B, $Q_{C,}$ and Q_D. The clock input is given to the input. As the pulse is given, the truth table is verified. Here the 8th pin is made high. The "1" value is shifted right by one position for every pulse. This is the circuit that can be used as a right shift register.

Right Shift Register

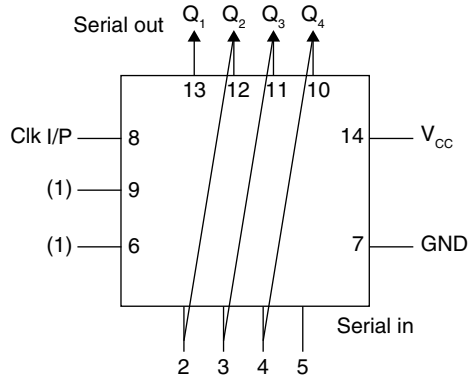


No. of pulses	Q_4	Q_3	Q_2	Q_1
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0
4	0	0	0	1

Left Shift Operation: To obtain the left shift operation again, the 14th pin is connected to V_{CC} and 7 to ground. The clock is given to the 8th pin. The pin 9 and 6 are made high. The output is got from 10, 11, 12, 13 pins. The pins 2 and 12 and 3 and 11, 14, and 10 are interconnected. The input is given to pin 5. 00 giving the clock input the

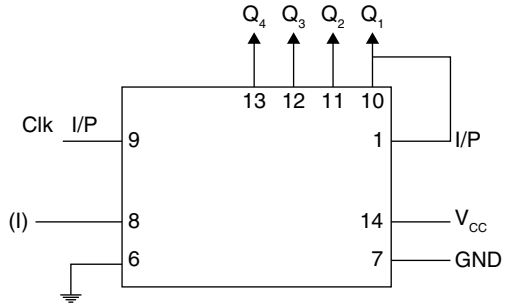
output obtained is verified with the truth here. Here the high value "1" level is shifted left for each pulse indicating the shift left operation.

Left Shift Register



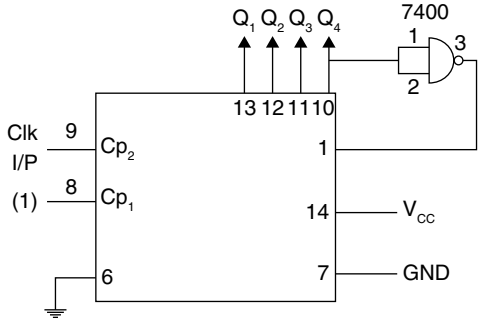
No. of pulses	Q_1	Q_2	Q_3	Q_4
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0
4	0	0	0	1

4-bit Ring Counter: The V_{CC} is connected to pin 14 and 7th pin is grounded. The 6th pin acts as mode control. The clock input is given to pin 9 and pin 8 is kept high. The output is taken from 13, 12, 11, and 10. The 10th and 1st pins are shorted.



No. of pulses	Q_4	Q_3	Q_2	Q_1
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0
5	0	1	0	0

4-bit Shift Counter: Here the pins 8 and 6 are made high. The clock is given to the 9th pin. Pin 10 and pin 1 are interconnected, through an inverter the outputs are verified when pulse is given.



No. of pulses	Q_4	Q_3	Q_2	Q_1
0	0	0	0	1
1	0	0	1	1
2	0	1	1	1
3	1	1	1	1
4	1	1	1	0
5	1	1	0	0
6	1	0	0	0

Result

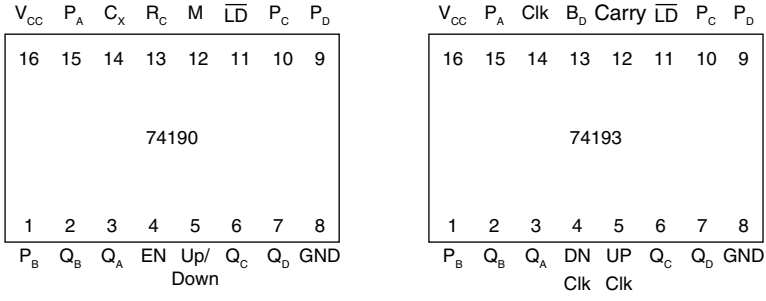
The shift left, shift right operations of a 4-bit shift register IC 7495 was studied and a MOD-4 ring counter and MOD-8 have been constructed.

STUDY OF UP/DOWN COUNTER

Aim

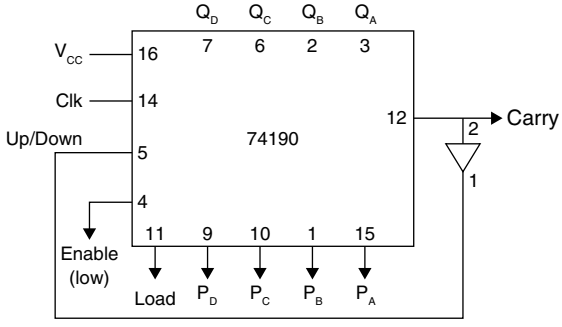
- To study the natural up/down counter sequences of IC 74190 and IC 74193 and the different modulus counts in the up and down sequence.
- Using the IC 74190 obtain the following binary equivalent decimal count sequence (a) 0123456789 87653210… (b) 456784321 45678…

Pin Configuration:

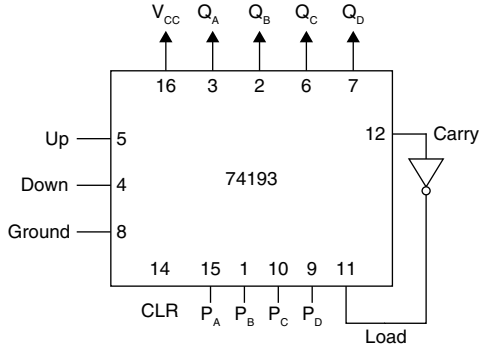

VCC PA CX RC M LD PC PD
16 15 14 13 12 11 10 9
74190
1 2 3 4 5 6 7 8
PB QB QA EN Up/ Down QC QD GND
VCC PA Clk BD Carry LD PC PD
16 15 14 13 12 11 10 9
74193
1 2 3 4 5 6 7 8
PB QB QA DN Clk UP Clk QC QD GND

Procedure

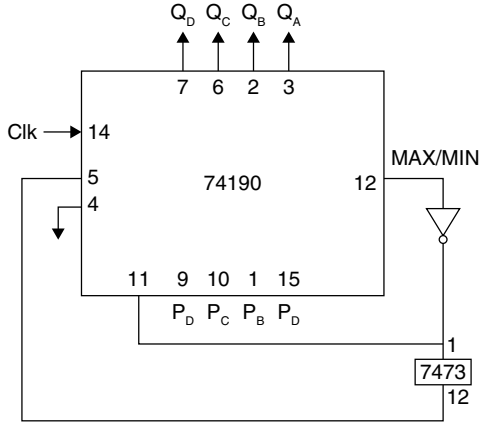
UP/DOWN Counter Using IC 74190: The circuit is wired up as shown in the circuit diagram below. In this, V_{CC} is connected to pin 10 and the clear terminal is connected to pin 14. The output from pin 12 is given to the input pin 2 of the inverter is given to 7407 chip. The pins 3, 4, and 14 are connected to V_{CC}. The inputs PA, PB, PC, PD are given to 7, 6, 2, and 3. By connecting pin 5, the IC 74190 can be allowed to count down made. The output from 7404 is given to pin 5 as input.



UP/DOWN Counter Using IC 74193: The circuit is wired up as shown in the circuit diagram. The V_{CC} is connected to pin 16. The clear terminals are grounded. The inputs are given through 15, 10, 1, 9, which are P_A, P_B, P_C and P_D. The ground terminal is connected to pin 5 and the down pin is connected to pin 4. The output pins Q_A, Q_B, Q_C, and Q_D and the pins 3, 2, 6, and 7. The pin 12 is connected to the input pin of the inverter and the output from the inverter is connected to the load of the IC 74193. This pin 12 is called the carry and pin 13 is the borrow.



Binary Equivalent Sequence Generator: The circuit shows that when pin 5 is connected to clock up and the pin 4 is connected to clock down. When the pin 4 is made high, the counter counts in a down made; and when the pin 5 is made high, the counter starts counting in a up mode. Thus, IC 74193 can be used as a MOD UP/DOWN counter.



Result

The natural UP/DOWN count sequences were studied. The binary equivalent decimal count sequences using 74190 was obtained.

DESIGN OF COUNTERS

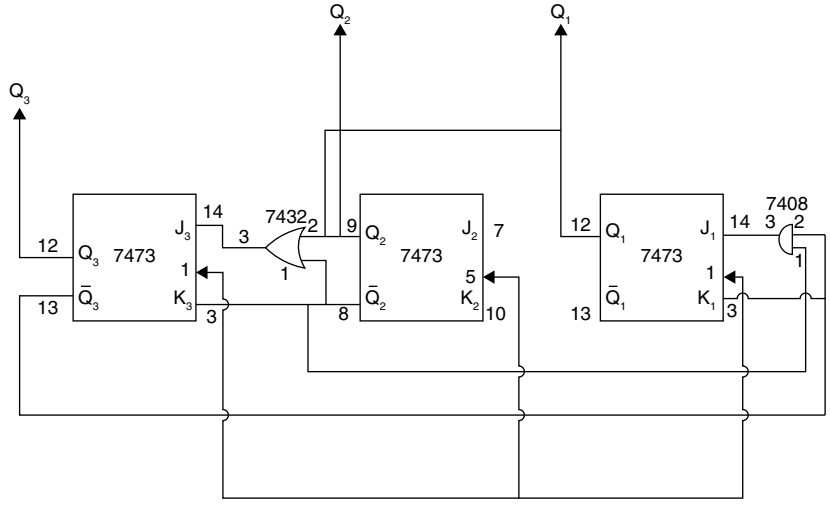
Aim

Design a counter using JK flip-flops and any other gates, which counts in the sequence given below:

State	Q_3	Q_2	Q_1
0	0	0	0
1	1	1	1
2	1	0	1
3	0	1	1
4	1	0	0
5	0	1	0
0	0	0	0

Procedure

Designing the Circuit: The experiment involves the designing (or) construction of the circuit or counter for the given sequence. The first step of designing includes the construction of the truth table and the execution table. The execution table gives the transition state for various conditions of J and K. For example, when J and K are 0 and do not care, the transition is from Q_n to Q_{n+1} and from 0 to 0. Similarly, when J and K are 1 and do not care, gives the transition from 0 to 1. Similarly, when J and K condition is do not care and 0, the transition is from 1 to 1. Using the truth table and the execution table, we construct the table for constructing the Karnaugh map.



The Karnaugh map is constructed using the truth table and the values for J_1K_1, J_2K_2, J_3K_3 are determined. On determining the values of the two JK flip-flops, the Ct is constructed. The working of the circuit can be verified. On checking the working of the circuit, the given sequence is obtained. If the given sequence is obtained then the circuit is assumed to the proper.

Result

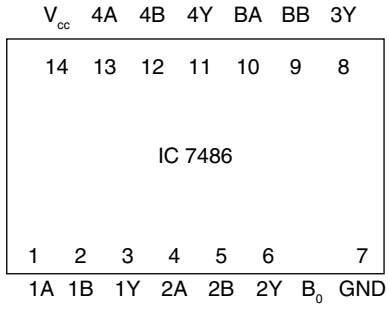
A counter using JK flip-flop of the given sequence was constructed and verified.

STUDY OF BINARY ADDER AND SUBTRACTOR CIRCUITS

Aim

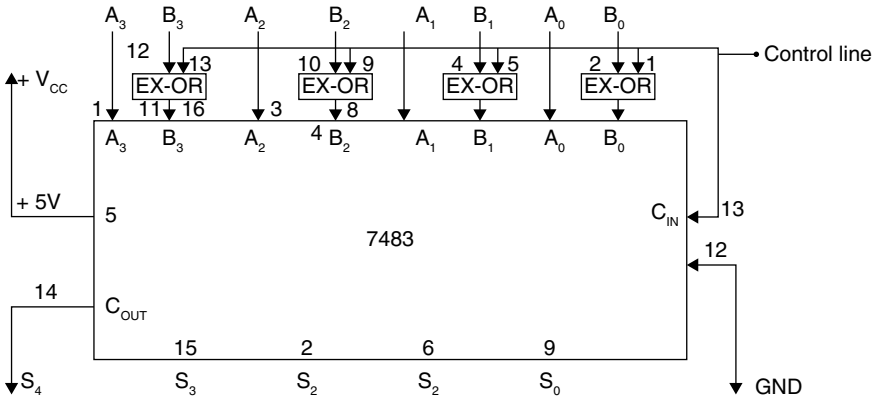
- Construct a 4-bit binary adder and subtractor with selection control using IC 7483 and 7486.
- Construct a full subtractor using IC 7483 employing 1st complement and 2s complement technique.

Pin Configuration:



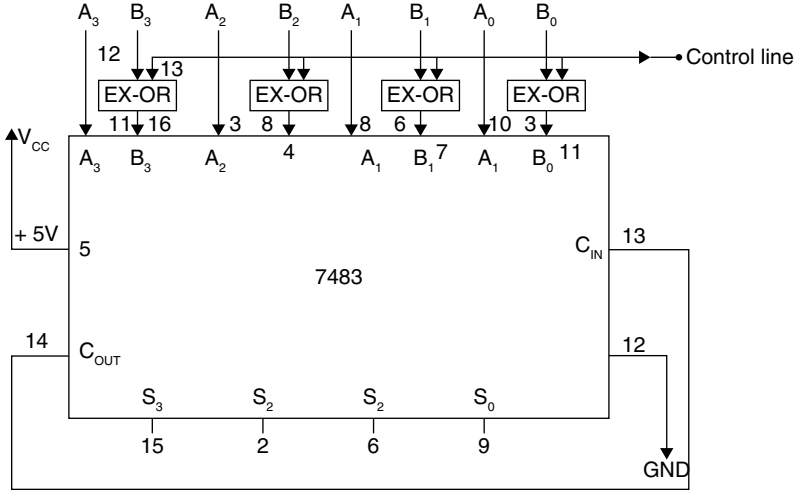
Procedure

Binary Adder Using IC 7483: The circuit is wired up as shown in the circuit diagram. The V_{CC} is given to pin 5 and pin 12 is grounded. The outputs S_0, S_1, S_2, S_3 is from pins 15, 2, 6, and 9. The inputs are given through A_0 to A_3 lines and other sets of inputs from B_0 to B_3 but through EX-OR gates. The control line is given to Ex-OR gate. The output from the Ex-OR gate is given as other input to the 7483 chip. The other connection from control line is given to C_{in}. The other pins are 1, 16, 3, 4, 8, 7, 10, and 11. On proving the required inputs, the output can be obtained, the output obtained will be the addition of 4-bit binary nor given through lines A_0 to A_3 and lines B_0 to B_3.



A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	S_4	S_3	S_2	S_1	S_0
0	0	0	1	0	0	0	1	0	0	0	1	0
1	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	0

Binary Subtractor Using IC 7483: The process involves the subtractor of the given 4-bit binary numbers using IC 7483. In this, the circuit is wired up in the following manner. Here again the two 4-bit binary numbers are given through lines A_0 to A_3 and other set of lines from B_0 to B_3 representing the second bit. The input through lines B_0 to B_3 are given through inverters as input. Here the C_{in} and C_{out} (i.e.) the carry is and carryout are connected together. By giving the two inputs, the two 4-bit binary numbers corresponding output is obtained across S_0, S_1, S_2, and S_3 with the pins 15, 2, 6, and 7. Thus, a binary output is achieved using a binary subtractor with 7483.



A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	D_3	D_2	D_1	D_0
1	1	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	0	0	0	0	1
1	1	1	1	1	1	0	0	0	0	1	1
1	1	1	1	1	0	0	0	0	0	1	1
1	1	1	1	0	0	0	0	1	0	1	1
1	0	0	0	1	0	0	0	0	0	0	0

Result

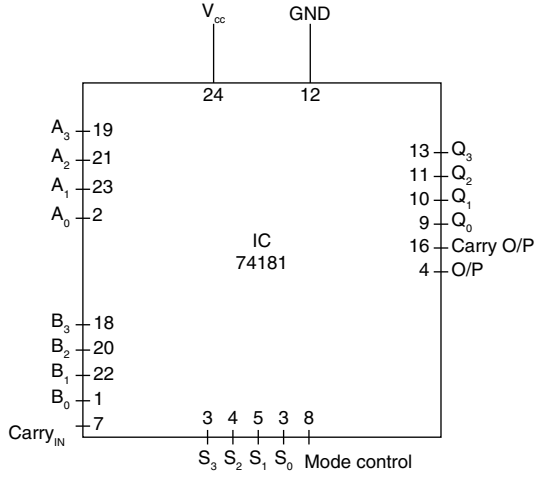
A 4-bit binary adder and subtractor with selection control using 7483 and 7486 were constructed and its working verified and subtractor using compliment technique was constructed and verified.

STUDY OF ALU AND BCD ADDER CIRCUITS

Aim

- To study the various logical operations and arithmetic operations using arithmetic and logic unit IC 74181
- Construct a BCD using IC 7483 and IC 7410 NAND GATE.

Pin Configuration:

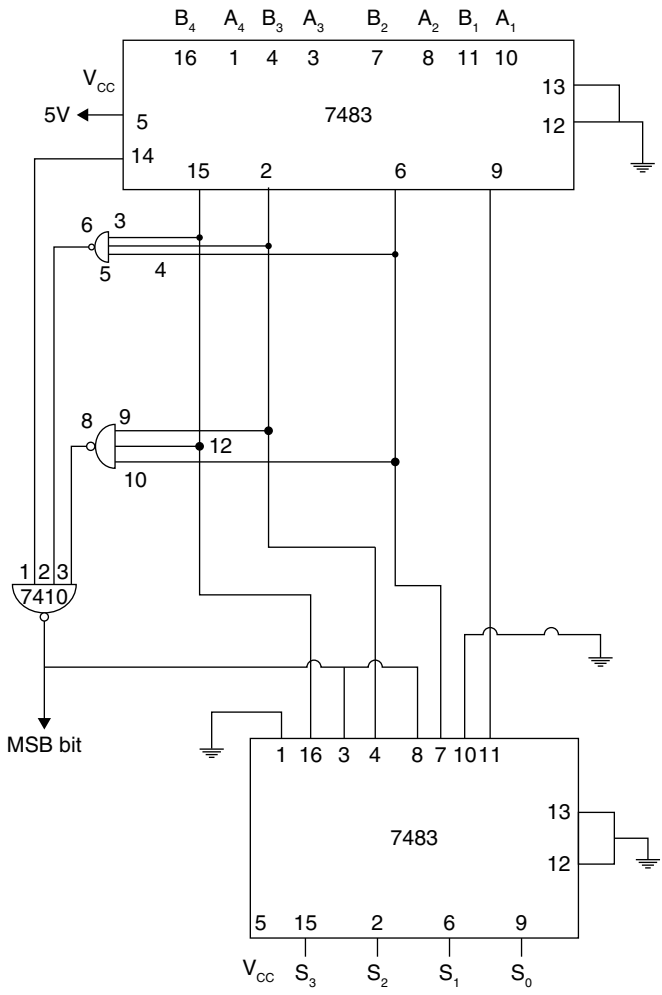


Procedure

The connections of the arithmetic and logic unit where IC 74181 are given and the following logic operation and arithmetic operations are checked.

Function separation				Logic M = 1	Arithmetic operations	
S_3	S_2	S_1	S_0	Operation	M = 0, C_n = 1	M = 0, C_n = 1
0	0	0	0	$F = \overline{A}$	F = A	A PLUS 1
0	0	0	1	$F = \overline{A + B}$	A + B	(A + B) P_1
0	0	1	0	$F = \overline{A}.B$	$A + \overline{B}$	$(A + \overline{B})$ P_1
0	0	1	1	F = 0	MINUS 1	0
0	1	0	0	$F = A.\overline{B}$	A PLUS $A\overline{B}$	A P_L $\overline{AB}$ P_L
0	1	0	1	$F = \overline{B}$	$(A + B) + A\overline{B}$	$(A + B$ PL $A\overline{B}$ P_L
0	1	1	0	$F = A \oplus B$	A MIN B MIN 1	A MIN B
0	1	1	1	$F = \overline{AB}$	AB MIN 1	$A\overline{B}$
1	0	0	0	$F = \overline{A} + B$	A PLUS AB	A P_L (AB) P_L
1	0	0	1	$F = A \oplus \overline{B}$	A PLUS B	A P_L B P_L 1
1	0	1	0	F = B	(A + B) PLUS B	$(A + \overline{B})$ P_L (AB) P_L
1	0	1	1	F = AB	AB MIN 1	AB
1	1	0	0	F = 1	A Plus A	A P_L A P_L 1
1	1	0	0	$F = A + \overline{B}$	(A + B) Plus A	(A + B) P_L A P_L
1	1	1	0	F = A + B	(A + B) Plus A	(A + B) P_L A P_L
1	1	1	1	F = 1	A minus 1	A

BCD Adder Circuits: This is useful in adding two BCD binary numbers. The circuit consists of two 7483 IC chips, 3-input NAND gates and an inverter. The circuit is wired up as shown in the circuit diagram. The inputs are given through S terminals. One of them through the inputs A_0, A_1, A_2, and A_3 and the other BCD one through B_0, B_1, B_2, and B_3. The outputs are obtained across the outputs S_0, S_1, S_2, and S_3. These 4 outputs are useful if the added 4-bit binary does not carry. If a carry is present then the 5th bit or the carry bit will be obtained across the terminal S_4. For example, when we add two BCD numbers 11 and 2, the resultant refers to 13 of the total output obtained across, and thus a 4-bit binary adder is obtained using 7483.



Result

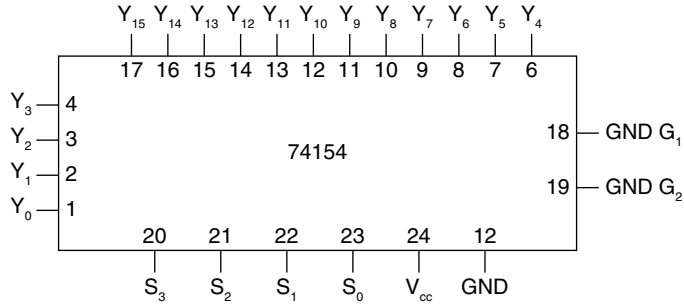
The binary coded adder using IC 7483 and 7410 gate were constructed and verified. The various logical and arithmetic operations using 74181 were verified.

STUDY OF DEMUX CIRCUITS

Aim

- To study the function of 74155 and 74154
- To construct 2–4 line decoder using NAND gates
- To realize a 3–8 DEMUX using 2-4 DEMUX

Pin Configuration:

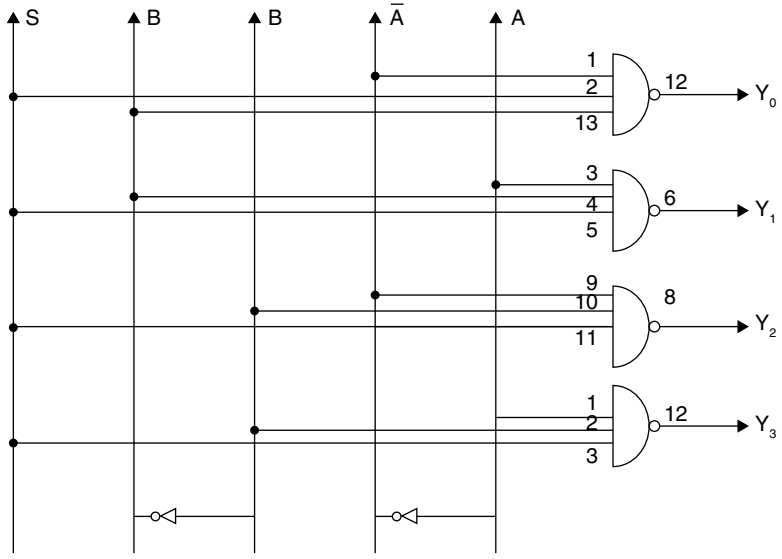


Procedure

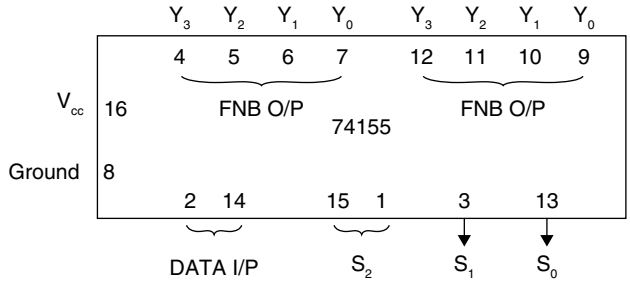
2–4 Line Decoder Using NAND Gates: The 2–4 line decoder circuit using NAND gates is shown in the circuit below. It consists of 4 7410 NAND gates. The 3-input NAND gates has 1 input from any of the 4 inputs and 3rd input is taken from the input 5. For the first NAND gate the inputs 1, 2, and 13. The output is obtained across pin 12. Similarly, the connections for the other NAND gates are also given. Thus, 4 outputs Y_0, Y_1, Y_2, and Y_3 are obtained and the obtained data are checked with the given table.

S_3	S_2	S_1	S_0	Data line selector
0	0	0	0	Y_0
0	0	0	1	Y_1
0	0	1	0	Y_2
0	0	1	1	Y_3
0	1	0	0	Y_4
0	1	0	1	Y_5
0	1	1	0	Y_6
0	1	1	1	Y_7
1	0	0	0	Y_8
1	0	0	1	Y_9
1	0	1	0	Y_{10}
1	0	1	1	Y_{11}
1	1	0	0	Y_{12}
1	1	0	1	Y_{13}
1	1	1	0	Y_{14}
1	1	1	1	Y_{15}

Functions of 74155 and 74154: The circuit is connected as shown in the circuit diagram. The AND and the V_{CC} is connected to 8 and 16 pins. The data select lines are from Y_0 to Y_4 and a data input line is given to connecting 2 and 14. The outputs are detained for S_0, S_1, and S_2.



3–8 Line DEMUX Using Dual 2–4 Line DEMUX: The circuit for 74155 has a circuit which is connected as shown in the circuit diagram and the truth table of the certain truth table is checked. Using the 74154 IC chip, the circuit diagram is connected using the circuit diagram. The truth table for 74154 is checked. Thus, the functions of 74155 and 74154 can be checked by the above-described manner.



S_2	S_1	S_0	Data line
0	0	0	Y_0
0	0	1	Y_1
0	1	0	Y_2
0	1	1	Y_3
1	0	0	Y_4
1	0	1	Y_5
1	1	0	Y_6
1	1	1	Y_7

Result

The 2–4 line decoder using NAND gates was verified and the functioning of 74155 and 74154 is constructed and verified.

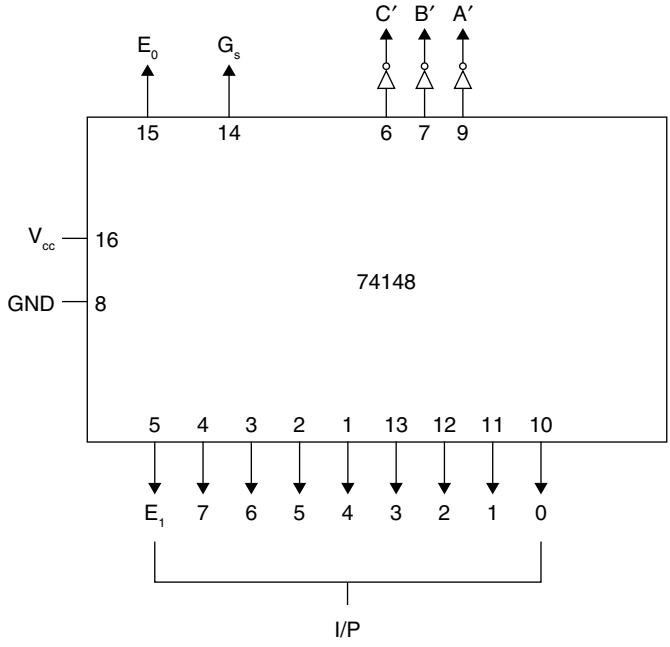
STUDY OF DIGITAL ENCODER/DECODER CIRCUITS

Aim

- To study the encoder circuits
- Octal to binary using IC 74148
- Hex to binary using IC 74148
- To study the decoder circuits
- BCD to decimal using IC 7442
- BCD to 7-segment decoder using IC 7447

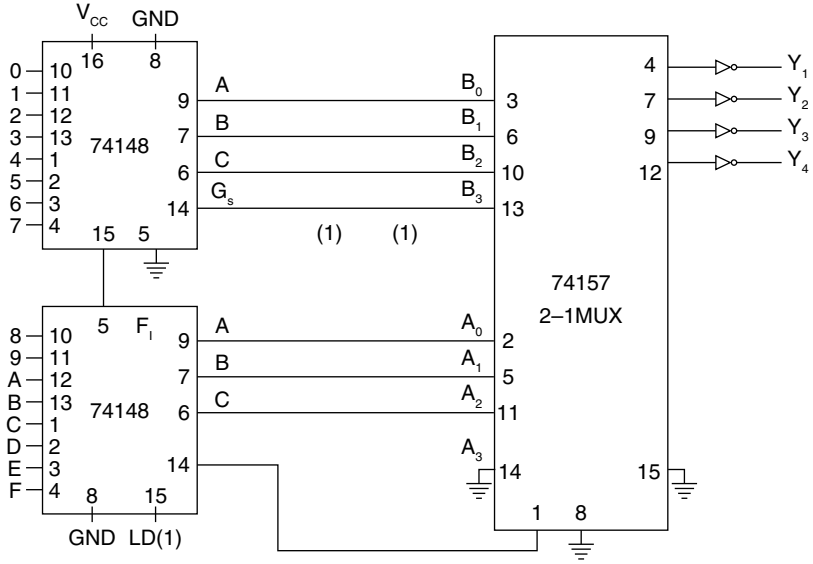
Procedure

Octal to Binary Using IC 74148: Wire up the circuit as shown in the figure, pin 16 is connected to V_{CC} and pin 8 is grounded. There are seven input lines and three output lines. When the enable line (pin 5) is logic "1", all the seven inputs are do not care. So the corresponding outputs are all zero, when the enable pin is logic "0". When the inputs are given, we get the corresponding binary output.

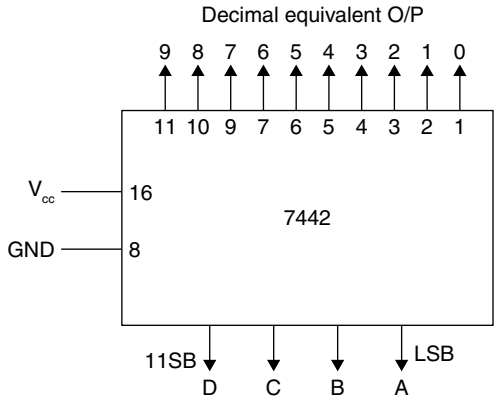


EI	0	1	2	3	4	5	6	7	G_s	E_0	C	B	A	C′	B′	A′
1	X	X	X	X	X	X	X	X	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0
0	X	0	1	1	1	1	1	1	0	1	1	1	0	0	0	1
0	X	X	0	1	1	1	1	1	0	1	1	0	1	0	1	0
0	X	X	X	0	1	1	1	1	0	1	1	0	0	0	1	1
0	X	X	X	X	0	1	1	1	0	1	0	1	1	1	0	0
0	X	X	X	X	X	0	1	1	0	1	0	1	0	1	0	1
0	X	X	X	X	X	X	0	1	0	1	0	0	1	1	1	0
0	X	X	X	X	X	X	X	0	0	1	0	0	0	1	1	1
0	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0

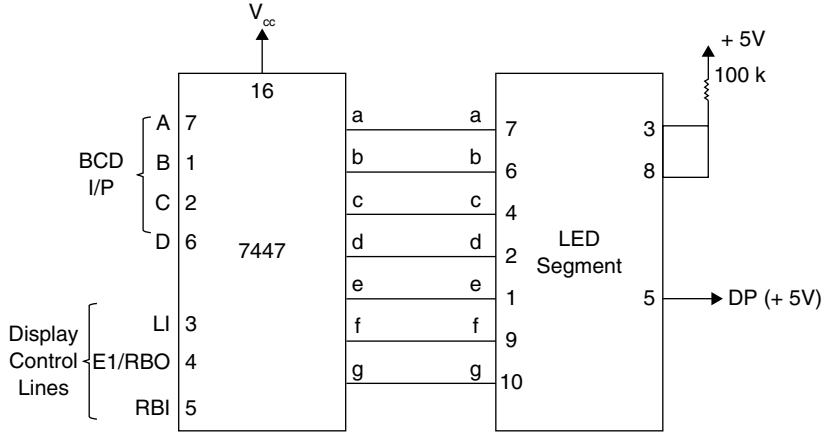
HEX to Binary Using 74148: Wire up the circuit as shown in the figure. The two ICs 74148 are interconnected by connecting pin 15 of the first IC to pin 5 of the second. Here, there are 16 hex input lines and four output lines. The output from both the ICs 74148 is given as input to 74157 which is 2–1 MUX. The output from 2–1 MUX is taken by connecting a not gate in between, when a hex value is given as input, the corresponding binary value is in the output line.



BCD to Decimal Using IC 7442: Wire up the circuit as shown. There are four input lines and eight output lines. Pin 16 is connected to V_{CC} and pin 8 to ground. When a particular BCD value is given as the input, the corresponding decimal value output goes low.



BCD to 7-Segment Decoder Using 7447: Wire up the circuit as shown. In the display control unit, when the lamp test LT is logic zero, all the segments glow. When BI/RBO is logic "1" all segments are off. When LT and RBT are logic "1" all segments normal decoding takes place when it is logic "0" normal decoding with zero blanking takes place. When a BCD number is given as input, the same number is display in 7-segment decoder.



Result

The encoder circuits of octal to binary and hex to binary using IC 74148 was studied and the decoder circuits of BCD to decimal and BCD to 7-segment decoder using IC 7447 were studied.

STUDY OF RAM

Aim

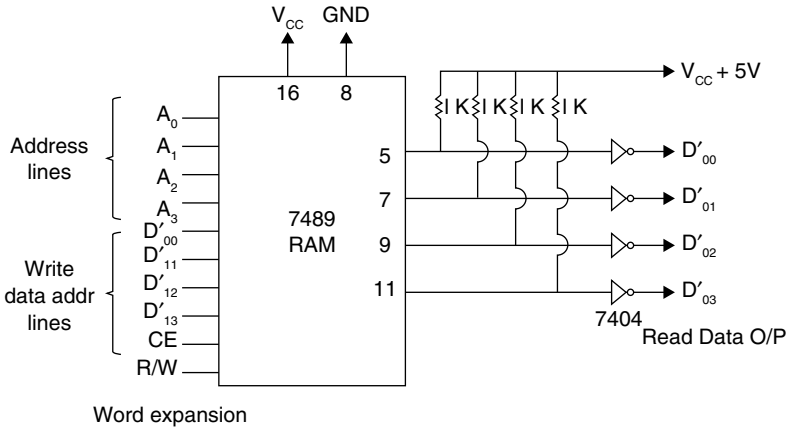
- To study the read/write operations of 16 × 4-bit RAM 7489
- Construct and verify the address (32 × 4) and word expansion circuits of 7489 (16 × 8).

Procedure

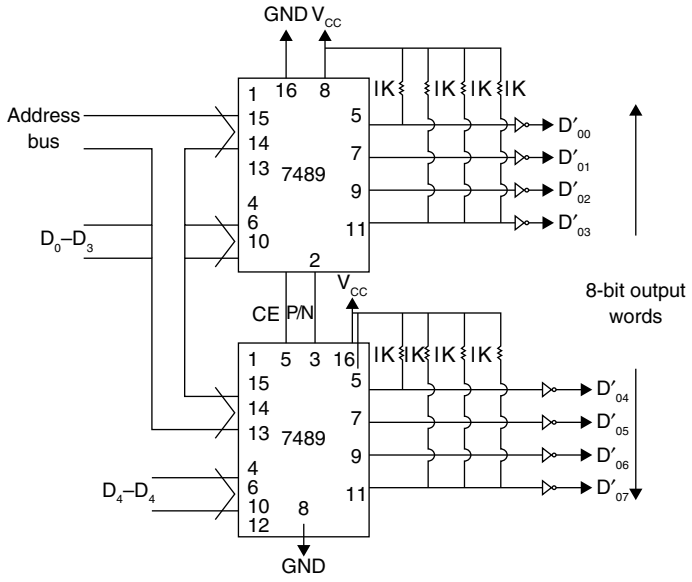
To Write: Apply the address of the memory location at the address input lines and apply data to be stored at the write data input lines and change the $\overline{CE}$ pin from logic "1" to logic "0". Then the data will be written into the memory location. But the outputs will be at all 1's at DO_0, DO_1, DO_2, DO_3 terminals. Then take the $\overline{CE}$ pin to logic "1" stored before writing another data. The process is repeated and the numbers of data are written into the memory location.

To Read: To read from any location, apply the address input lines and apply logic 1 at $R/\overline{W}$ line and then change $\overline{CE}$ from logic "1" to logic "0" and the data is read take back $\overline{CE}$ pin to logic "1" before changing the address to read from other locations.

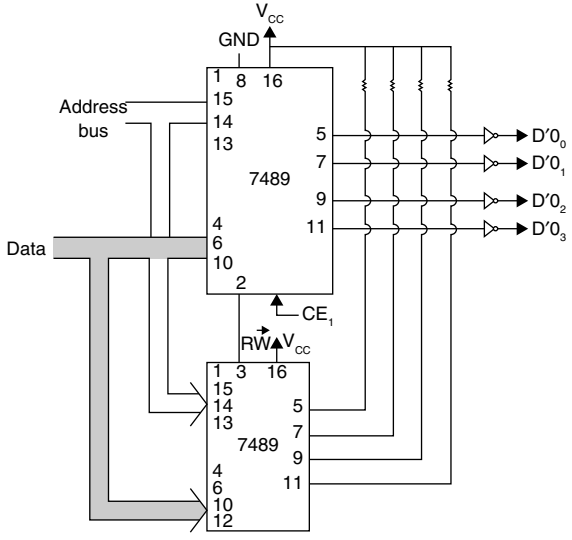
READ/WRITE Operations of 16 × 4-bit RAM 7489

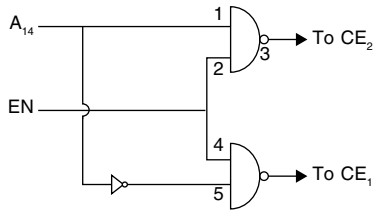


Word Expansion: (16 × 8) RAM (7489) 16 words of 8-bit each. Here two chips of 7489 are used and all their inputs are interconnected to give all expansion. Now give the address which is four but each and the word are eight but each.



Address Expansion: 32 × 4-bit RAM (7489) 32 words of 4-bit each. Here we use two NAND gates to enable each of 7489 chip. A_4 line is given as one input of NAND gate to which an inverter is connected and given as the other input of NAND gate. A common switch is taken as the enable line and given to either of NAND gates as input. When this enable line is zero, both the chips are disabled but both line A_4 will enabled. When any logic is given to line A_4 will disable one chip and enables the other as an inverter is connected to it. Now all the address lines and data lines of each chip is independent word we can get 32 words of 4 but each.





Result

The read/write operations of 16 × 4-bit RAM 7489 is studied. Word expansion and address expansions were constructed and verified.

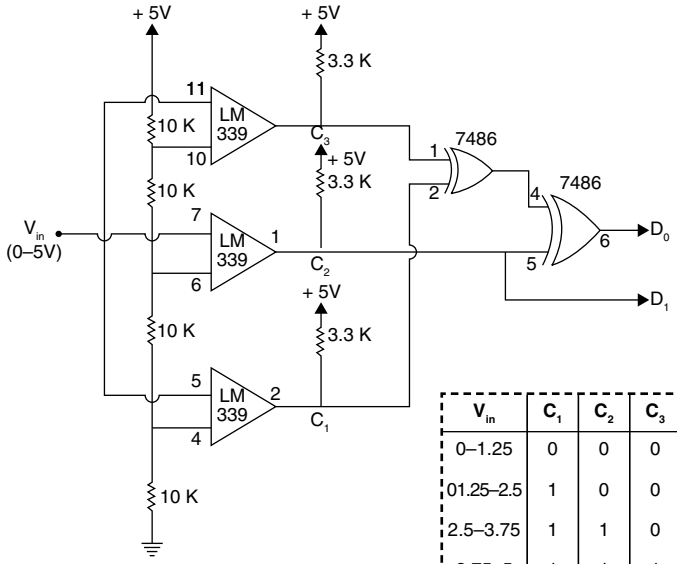
A/D CONVERTER

Aim

To construct (i) a 2-bit and (ii) 3-bit simultaneous analog to digital convertor using LM 339 and the necessary encoding and verify the operations.

Procedure

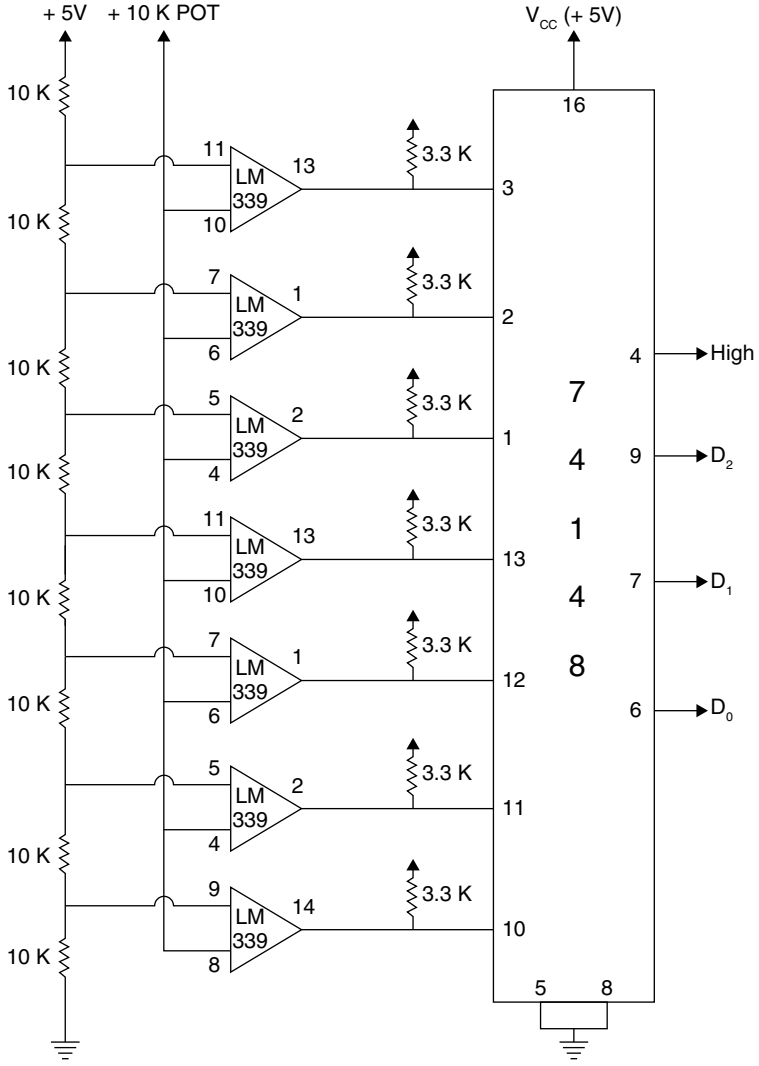
2-bit Simultaneous A/D Converter: The circuit is constructed as shown in the circuit diagram. This type is the fastest and simplest form of A/D converter. The 2-bit A/D converter uses the comparators. A voltage of about 5V is given which is shared



V_{in}	C_1	C_2	C_3	D_1	D_0
0–1.25	0	0	0	0	0
01.25–2.5	1	0	0	0	1
2.5–3.75	1	1	0	1	0
3.75–5	1	1	1	1	1

between each comparator. Due to resistor 10K, there is drop in voltage across each comparator. The inputs of all the comparators are shorted together and a voltage is given. But because of the resistors, the voltage across each comparator varies and as a result, the applied voltage is 5V. When the voltage is set between 0 and 1.25V, we have the corresponding comparator O/P if the voltage lies between 1.25 and 2.50V, we have its corresponding output and thus the voltage is varied up to 5V, the outputs are obtained according to this. These outputs are obtained from the comparator are given to the EX-OR gates to get the digital output.

3-bit A/D Converter: The same procedure for 2-bit A/D converted is repeated, but the output of the comparator is given to chip 74148 which gives the digital output.



Result

2- and 3-bit A/D converter using LM339 were studied and verified.

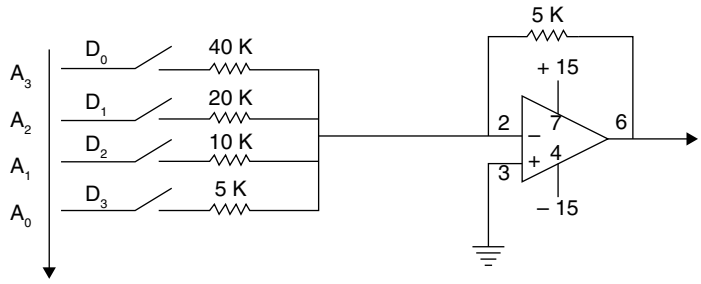
D/A CONVERTER

Aim

- To construct a weighted D/A converter
- To construct a 8-bit DAC using 0808

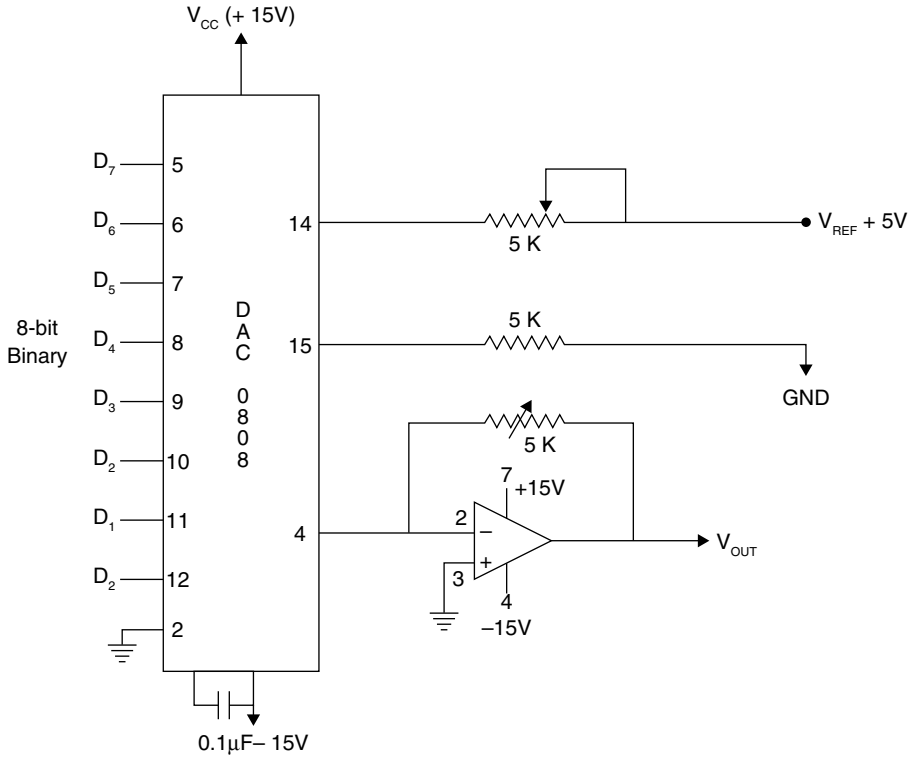
Procedure

4-bit Binary Weighted Resistor: The circuit is wired up as shown in the circuit diagrams. As the data inputs are varied due to each resistor, the voltage increases showing a linearity between the digital input and analog output. As the digital output is increased, the analog voltage is obtained.



D_0	D_1	D_2	D_3	V_{OUT}
0	0	0	0	0
0	0	0	1	0.29
0	0	1	0	0.89
0	1	0	0	1.15
1	0	0	0	1.40
1	1	1	1	1.45

8-bit DAC Using 0808: DAC 0808 is a 14 pin chip with the data inputs given to the corresponding input pins. It functions as mentioned above. The digital input is varied according to which the analog voltage is calculated. We observe that as the digital input is incremented stepwise, there is a linearity graph obtained between the digital input as the analog output voltage for various data input lies the analog voltage is calculated in the counter and then displayed.



$$V_o = \left(\frac{V_{REF}}{R_{14}}\right) \cdot R_0 \left\{\frac{D_7}{2} + \frac{D_6}{4} + \frac{D_5}{8} + \frac{D_4}{16} + \frac{D_3}{32} + \frac{D_2}{64} + \frac{D_1}{128} + \frac{D_0}{256}\right\}$$

$$V_o = \frac{5}{2k} \cdot 2k \cdot \left\{\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256}\right\}$$

$$= 5 \times \frac{225}{256} = 4.98\,V$$

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1

Result

A weighted D/A convertor was studied and verified. Also 8-bit DAC using DAC 0808 was constructed.

STUDY OF MUX

Aim

To construct

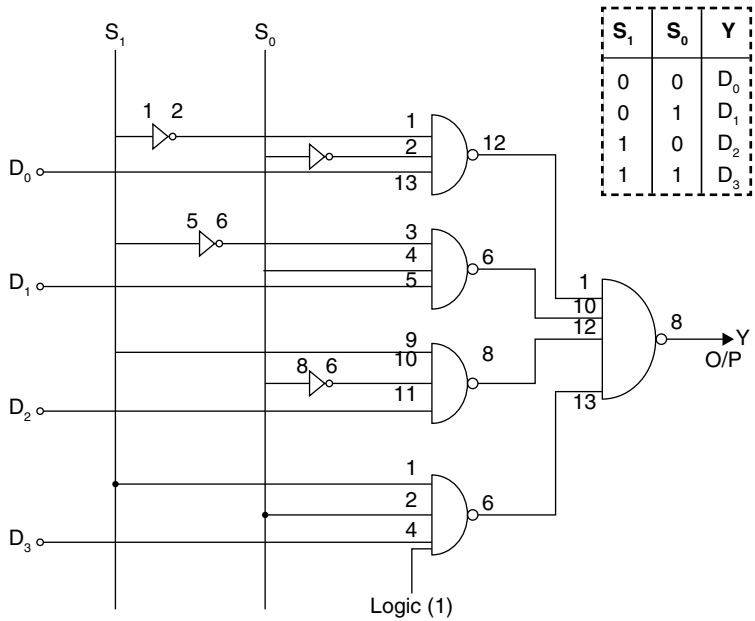
- 4-1 multiplexer using NAND gates
- To study the function of 74157, 74153 MUX
- To realize 8–1 MUX using 4–1 MUX (74153)
- To realize a given logic expression using MUX

$$y = \bar{A}B\bar{C} + A\bar{B}\bar{C} + \bar{A}BC + ABC$$

Procedure

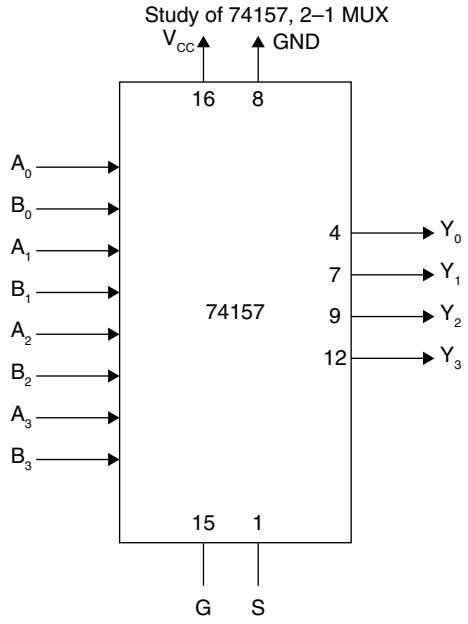
The NAND gate in the circuit functions as a 4–1 multiplexer. We require 4 input lines and one output line. According to the input data, the required output line is selected.

4–1 MUX Using NAND Gates:



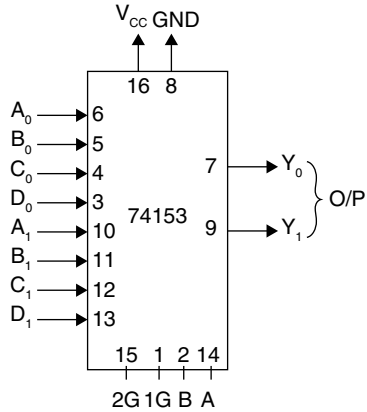
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Function of 74157: It functions as a 2–1 multiplexer. It has eight input lines and four output lines. Pin 10 is connected to V_{CC} and pin 8 is grounded. Pin 1 is the select line pin 15 is the strobe. According to the values in the strobe and select, the value in the input line is transmitted to the corresponding output line. When the strobe is zero and select line is do not care all the outputs become zero. We get the output only when the strobe is zero.

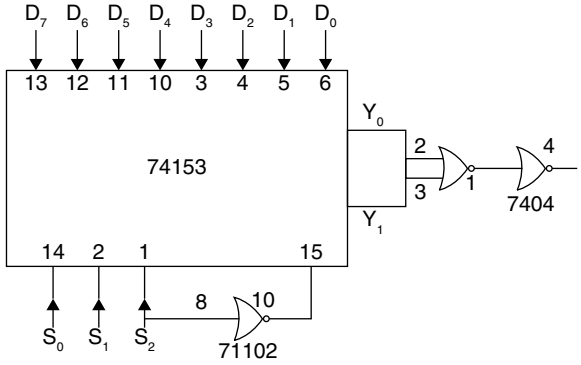


Input		Output			
Stroke G	Stroke S	Y_3	Y_2	Y_1	Y_0
0	0	A_3	A_2	A_1	A_0
0	1	B_3	B_2	B_1	B_0
1	x	0	0	0	0

Functions of 74153: It functions as a 4–1 multiplexes, it has 4 inputs and 2 outputs. Pin 16 is connected to V_{CC} and pin 8 is grounded. According to the values in the select lines S_1 and S_0, the value in the input lines is transmitted to the output. If the strobe is one, all are zero.

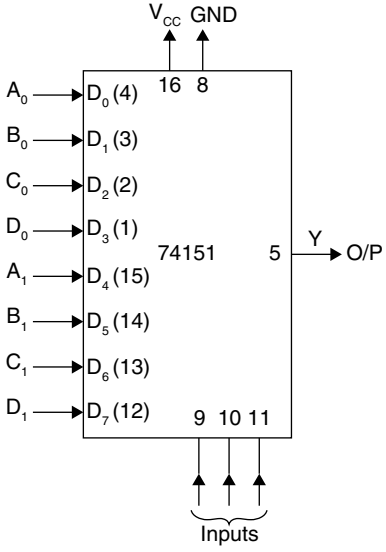


Select I/P		Output	
S_1	S_0	Y_0	Y_1
0	0	A_0	A_1
0	1	B_0	B_1
1	0	C_0	C_1
1	1	D_0	D_1



S_2	S_0	S_0	Data enable
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

8–1 MUX Using Dual 4–1 MUX: It has three input lines and eight output lines. We require three controls to select one of the 16 output lines. According to the input data in the three control lines, the respective output line is selected using. This 74153 in the given logical expression is verified.



$$Y = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}BC + ABC$$

C	B	A	Output Y	Inputs
0	0	0	0	D_0
0	0	1	1	D_1
0	1	0	1	D_2
0	1	1	0	D_3
1	0	0	0	D_4
1	0	1	0	D_5
1	1	0	1	D_6
1	1	1	1	D_7

Result

A 4–1 multiplexer using NAND gates was constructed. Functions of 74157, 74153 MUX were studied. The 8–1 MUX using 4–1 MUX was realized.

INDEX

2-bit PISO Register 238
2-bit Simultaneous A/D Converter 415
2-input NOR Gate 53
2–4 Line Decoder Using NAND Gates 408
256K × 8 Memory 286
3-bit A/D Converter 416
3-input NAND Gate 52
3–8 Line DEMUX Using Dual 2–4 Line DEMUX 409
4-bit Binary Weighted Resistor 417
4-bit Ring Counter 399
4-bit Shift Counter 399
4-1 MUX Using NAND Gates 419
7-segment Display 314
  common cathode and common anode LED displays 315
  construction and working 314
74LS165 8-bit Parallel Load Shift Register 239
8-bit DAC Using 0808 417
8-1 MUX Using Dual 4–1 MUX 421

A

Access Time 282
AC Clocking 200
Accuracy of A/D Conversion 335
Accuracy of D/A Conversion 345
Active Low SR Latch 194
Active Matrix 309
ADC Process 334
  encoding 335
  quantizing and encoding (Q/E) 334
  resolution 335, 345
  sampling and holding (S/H) 334
Advantages and Disadvantages of Digital Signal Transmission 5
ALGOL 381
Alphabetical 372
Alphanumeric Codes 30
  American Standard Code for Information Interchange (ASCII) 30
  Extended Binary Coded Decimal Interchange Code (EBCDIC) 31
  Hollerith code 30
Analog Computers 360
Analog Signals 1
Analog-to-digital Convertor 331
AND Gate 46
  symbol for AND Gate 47
Applications of DAC 351
Applications of Flip-flops 216
  frequency division 216
  parallel data storage 216
Arithmetic Logic Unit (ALU) 365
Artificial Intelligence 360
Assembler 382
Assembly Language 379
Astable Multivibrator 221, 387
Asynchronous Sequential Circuits 190, 191
  address expansion 414

B

Back-lit LCD 305
BASIC 381
BCD Adder Circuits 406
BCD Counter 262
  two-digit BCD counter 263
BCD to 7-segment Decoder Using 7447 412
BCD to Decimal Using IC 7442 411
Bidirectional Shift Registers 242
Binary Adders 163
Binary Adder Using IC 7483 403
Binary Addition 15
Binary Division 22

Binary Equivalent Sequence Generator 401
Binary Multiplication 21
Binary Numbering System 10
Binary Subtraction 17
Binary Subtractor Using IC 7483 404
Binary to Hexadecimal 15
Binary to Octal 14
Binary-weighted Resistor DAC 346
working 347
Bipolar Digital Signal 2
Bipolar Transistors 140
construction, symbol and nomenclature 140
current amplification factor 141
the active linear region 142
the cut-off region 142
the saturation region 142
working of NPN BJT transistor 140
Bistable Multivibrator 217
Block Diagram of DRAM 288
Block Diagram of RAM 282
Block Symbol of 8-bit SISO 235
Boolean Algebra 73
Boolean Algebraic Properties 73
associative property 74
commutative property 74
distributive property 75
Burst EDO DRAM (BEDO DRAM) 291

C

Cache Hits 366
Cache Memory 365
Cache Miss 366
Carry Look Ahead Counter 267
Cathode Ray Tubes (CRTs) 301
CD-E 371
CD-R 371
CD-ROM 366, 369, 370
Central Processing Unit (CPU) 365
Chord and Numeric 372
Circuit and Symbol of JK Flip-flop 204
C Language 382
Classification of Computers 360
Classification of ICs 132
Clocked SR Latch 195
clocked SR using NAND gates 196
Clock Pulses 197
CMOS 154
CMOS Gate 385
CMOS Inverter 154
CMOS NAND Gate 155
CMOS Protection 156
COBOL 380
Code Convertors 180
BCD to 7-segment decoder 180
Colour CRTs 302
Colour LCD 308
Column Address Strobe (/CAS) 289
Combinational Circuits 163, 189
Common Digital IC Terminology 132
Comparison of TTL and CMOS Logic Families 156
Compilers 382
Complementary Metal-oxide Semiconductor 153
Computer Characteristics 355
Computer Software 377
application software 378
Construction of LCD Display 306
Contact Bounce Eliminator 195
Control Unit (CU) 365
instruction pointer 365
instruction registers 365
Conversion of Numbers 12
binary to decimal 12
binary to hexadecimal 15
binary to octal 14
decimal to binary 12
decimal to hexadecimal 14
decimal to octal 13
hexadecimal to decimal 14
octal to binary 14
Conversions of Flip-flops 210
Counters 257
2-bit counter 257
asynchronous counter 260
synchronous counter 265
Customized Softwares 378
Cycle Time 282

D

Decade Counter or MOD-10 Ripple Counter 262
Decade Synchronous Counter 267
Decimal Numbering System 10

Decimal to Hexadecimal 14
Decoders 181
Decoding a Counter 264
De Morgan's Theorem 75
Demultiplexer 177
1-to-4 line DEMUX 179
2-to-4 decoder 178, 181
3-to-8 decoder 183
IC 74LS138 179
Depletion Layer 137
Designing of Ripple Counter Divided by N 261
Design of Counters 402
D Flip-flop to JK Flip-flop 214
D Flip-flop to SR Flip-flop 213
D Flip-flop to T Flip-flop 215
Digital Computers 361
Digital Light Processing 323
Digital Micromirror Device (DMD) 323
working of 325
advantages of 325
Digital Ramp ADC 336
working 336
Digital Signals 1
Digital-to-analog Convertors 344
characteristics of 345
Digital Transmission 3
Diode Logic 144
disadvantages of diode logic 146
DL AND gate 145
DL OR gate 145
Diodes 137
Diode–Transistor Logic (DTL) 148
DTL NAND gate 148
Direct Polarity Indication 58
Direct Recombination 312
Dirty Data 366
Disadvantages 310
Display Devices 301
Display Technology 301
D Latch 200
Don't Care 211
Don't Care Conditions 106
D (or) Delay–Direct Latch 192
Down Counters 261
3-bit or MOD-8 down counter 261
Dual In-line Memory Module (DIMM) 290
Dual In-line Package (DIP) 290
Dual Slope Convertor 343
Dvorak 372
Dynamic Power Dissipation (P_d) 135
Dynamic Random Access Memory (DRAM) 287
DRAM memory cell 287
DRAM memory modules 290

E

E-books/E-ink 325
Edge-triggered D Flip-flop 202
Electrically Triggered Start/Stop Gate 387
Emitter-coupled Logic 152
Encoders 184
8-to-3 encoder 185
End Around Carry (EAC) 19
Equivalent Gates 123
Error Detection 33
cyclic redundancy check code 36
even parity 34
Hamming code 38
Hamming distance 38
odd parity 34
parity bits 34
parity method 33
repetition code 35
Error Detection and Error Correction in Digital Transmission 4
Essential Prime Implicants 91
Even Modulus Shift Counter 389
Evolution of Computers 356
Exclusive-NOR Gates 55
EX-NOR Gate 56
Extended Data out DRAM (EDO DRAM) 291
Extrinsic Semiconductor 136

F

Fall Time 3
Fan-in 133
Fan-out 133
Fast Page Mode DRAM (FPM DRAM) 291
Feedback Loop 191
Field-effect Transistor (FET) 142
symbol of 142
junction field effect transistor 143
Fifth Generation 360
First Generation 358

Flags 231
Flash A/D Converter 340
flash ADC circuit 341
Flash Memory 293
Flip-flops 191
Floppy or Flexible Disks 367
Fortran 379
Fourth Generation 359
Frame Buffers 303
Full Adder 165
full adder circuit 166
full adder circuit using half adder 167
full adder using two HA 167
multi-bit addition using full adder 168
Carry Look-Ahead adder (CLA adder) 169
full adder circuit for operand bits addition 169
look-ahead carry adder 169
single-bit full adder 167
Full Scale Voltage (FSV) 345
Fundamental Symbols—Boolean Algebra 73
equal sign 73
inversion sign 73
multiplication sign 73
plus sign 73

G

Gas Plasma Displays 321
working of 322
types of 322
DC plasma display 323
AC plasma display 323
Generalized Packages 378
General Purpose Computers 362
Generations of Computers 357
Giant Scale Integration (GSI) 132
Glitches 346

H

Half Adder 164
Hard Disks 367
Hexadecimal Addition 16
Hexadecimal Numbering System 11
Hex to Binary Using 74148 411
High Level Language 379
Hybrid Computers 361

I

IC 131
IC 555 Timer 224
IC Counters 269
IC Packaging 157
IEEE Symbols 58
positive logic convention 58
negative logic convention 58
Implicant 91
Implementation of Boolean Functions using Multiplexers 175
combinational circuit implementation using multiplexers 176
Indirect Recombination 313
Input–Output Devices 371
Integrated Circuits 131, 359
Interpreters 382
Intrinsic Semiconductor 136
Inverter 49
Isotropic Liquids 304

J

Jack Kilby 131
JK Flip-flop 203
JK Flip-flop Data Transfer 209
JK Flip-flop to D Flip-flop 214
JK Flip-flop to SR Flip-flop 211
JK Flip-flop to T Flip-flop 213
JK Flip-flop Using NAND Gates 204
Johnson Counter 246
5-bit Johnson counter 246
Junction Diode 137

K

Karnaugh Map 91
Karnaugh Map for Five Variables 94
Karnaugh Map for Four Variables 93
Karnaugh map for JK Flip-flop 206
Karnaugh Map for Three Variables 93
Karnaugh Map for Two Variables 91
Keyboard 371

L

Laptops 364
Large Scale Integration (LSI) 132
Latch 192

SR latch 192
SR latch using NOR gates 193
Left Shift Operation 398
Left Shift Register 399
Light Emitting Diodes (LED) 312
advantages of 313
disadvantages of 313
LED display 314
Linearity 346
Liquid Crystal Displays (LCD) 304
advantages of 310
classification of 304
disadvantages of 310
types of 306
Liquid Crystals on Silicon (LCOS) 310
Logic Axioms 76
multivariate theorems 79
single variant theorems 77
Logic Diagram of 8-bit SIPO 237
Logic Families 144
Logic Gate 45
Logic Gate Delay 134
Logic Probe 158
Logic Pulser 158
Logic Symbol of 8-bit SIPO 237
Low Level Language 378

M

Machine Language 379
Magnetic Bubble Memory 369
Magnetic Disks 366
Magnetic Ink Character Reader (MICR) 374
Magneto Resistive RAM (MRAM) 294
Magnetic Tapes 369
Mainframe Computers 363
Map Rolling 98
Master–Slave D Flip-flop 202
Master–Slave JK Flip-flop 208
Maurice Karnaugh 91
Maximum Clock Frequency 267
MDAC Multiplying DAC 346
Medium Scale Integration (MSI) 132
Memory Devices 281
Memory Unit (MU) 365
Metal Oxide Semiconductors (MOS) 143
Microchip 131
Microcomputers 362
Microprocessors 359
Minicomputers 362
Minimal Sum of Products (MSP) 100
Minterms and Maxterms 79
maxterms 80
minterms 80
Minterm 91
Mixed-signal Device 333
MOD-3 Counter 258
MOD-5 Synchronous Counter 394
MOD-6 Synchronous Counter 394
MOD-7 Synchronous Counter 394
MOD-16 or MOD-4 Bit Counter 259
Modulus Counter – Decade Counter (7490) 395
Monitor 375
Monostable Multivibrator 219
Monotonicity 345
Most Significant Bits (MSB) 19
Mouse 372
barcode reader 374
cordless mouse 373
joystick 373
light pen 374
mechanical mouse 373
optical mouse 373
touch screens 373
trackball 373
Multiplexers 172
2 × 1 multiplexer 173
4 × 1 multiplexer 173
16 × 1 multiplexer 175
QUAD 2 × 1 multiplexer 174
Multivibrators 217

N

NAND 50
NAND-AND and AND-NOR Implementation 127
NAND and NOR to Boolean functions 124
NAND Implementation 121
Negative-AND Equivalent Operation of the NOR Gate 55
Negative Binary Numbers 18
Negative-OR 53
Negative OR-equivalent Operation of a NAND Gate 52
N-MOS 143
Noise Immunity and Noise Margin 133

NOR 50
NOR as a Universal Gate 122
NOR Gate 53
NOR Gate – Universal Gate 385
NOR-OR Implementation and OR-NAND Implementation 127
NOT Gate 49
  symbol for 50
N-type 136
Number Codes 23
  2421 code 24
  5211 code 24
  8421 code 24
  advantages of BCD codes 25
  advantages of the gray code 29
  applications of the gray code 29
  Binary Coded Decimal (BCD) or 8421 code 24
  conversion of binary to gray code 27
  conversion of gray code to binary 28
  CRC 36
  disadvantages of BCD codes 25
  excess-3 codes 25
  gray code 27
  limitations of the gray code 29
  non-Boolean gray code 29
  weighted codes 24
  XS-3 code 26
Number Systems 9

O

Octal Numbering System 10
Octal to Binary 14
Octal to Binary Using IC 74148 410
Octet 97
Odd Modulus Shift Counter 390
Offset Error 346
One's Complement 18
Operation of a 2-input NAND Gate 51
Operation of a NAND Gate 51
Operation of JK Flip-flop 204
Operation of Xnor Gate 57
Optical Code Reader (OCR) 374
Optical Diskettes 371
Optical Storage Systems 369
OR Gate 48
  symbol for 48
Organic Light Emitting Diodes (OLED) 317
  introduction to 317
  electroluminescence 317
  Small Molecules Organic Light Emitting Diodes (SMOLED) 317
  Polymer Light Emitting Diodes (PLED) 317
  components of 318
  construction of 318
  vacuum deposition or Vacuum Thermal Evaporation (VTE) 319
  inkjet technique 319
  working of 319
  active matrix displays (AMOLED) 320
  passive matrix displays (PMOLED) 320
  types of 320
Output Devices 374
Output Enable (/OE) 289
Overlapping Groups 98

P

Pairs 94
Palmtops 364
Parallel Adder–Subractor 171
  adder–subractor circuit 171
  binary multiplier 172
Parallel Counter (or) Synchronous Counter 265
  carry look ahead counter 267
  maximum clock frequency 267
Parallel In/Parallel Out Shift Register (PIPO) 241
Parallel In/Serial Out (PISO) Shift Register 238
Pascal 382
Passive Matrix 309
Personal Computers 364
Plotter 376
p-MOS 143
PN Junction Diodes 137
  voltage-current characteristic of diode 138
  working of PN diode 138
Polarizing Filter 305
Positive 18
Positive and Negative Logic 45
Power Dissipation 135

Prime Implicants 91
Principle of ADC 333
Printer 375
dot matrix printers 375
electrographic printer 376
electro static printer 376
inkjet printers 376
laser printers 376
Product of Sum Form (POS) 103
Programmable Array Logic 294
Programmable Logic Array 296
Propagation Delay 194
P-type Semiconductors 136
Pulse Disabling Gate 387
Pulse Enabling Gate 387

Q

Quads 96
QWERTY 372

R

Racing Condition 197
Random Access Memory (RAM) 281
Read-only Memory 291
Electrically Erasable PROM (EEPROM) 293
EPROM and EEPROM 293
programmable read only memory 292
ROM cell 291
Read Only Memory (ROM) 281
Recombination 137
Redundant Groups 98
Registers 231
Resistor Ladder D/A Converter 349
Resistor–Transistor Logic (RTL) 147
advantages and limitations 148
RTL inverter or NOT gate 147
Right Shift Register 398
Ring Counters 244
3-stage ring counter 245
circulating shift register 245
Rise Time 3
Row Address Strobe (/RAS) 289
RS to D Flip-flop 212
RS to T Flip-flop 215
Rules in Karnaugh Mapping 101

S

Schmitt Trigger 223
SDRAM (Synchronous DRAM) and DDR SDRAM (Double-Data Rate DRAM) 291
Second Generation 358
Semiconductors 135
Semiconductor Transistors 139
Sequence Generator 249
sequence generator circuit 250
Sequential 163
Sequential Circuits 189
Serial In/Parallel Out (SIPO) Shift Register 236
Serial In/Serial Out (SISO) 232
Serial to Parallel Conversion or Vice Versa 251
Servers and Clients 364
Setting Time 346
Set up and Hold Time 282
Shift Registers 231
4-stage shift register 233
applications of 244
classification of 232
Sigma-delta A/D Converter 342
Single In-line Memory Module (SIMM) 290
Single In-line Pin Package (SIPP) 289
Single-signal Devices 333
Small Scale Integration (SSI) 132
Snarf 366
Snoop 366
Specific Purpose Computers 362
Speech Input Unit 376
SR Flip-flop 200
SR Flip-flop to JK Flip-flop 210
SR (or) Set–Reset Latch 192
Stale Data 366
Standard NAND Gate Logic Symbols 51
State Diagram 206
State Diagram of Asynchronous Counters 260
State Diagram of SR Latch 195
Static Power Dissipation (P_s) 135
Static Random Access Memory (SRAM) 283
6-transistor CMOS SRAM cell 284
4 × 1 RAM 284

4 × 4 RAM 284
SRAM chip 283
Steering Gates 196
Strobing 264
Study of UP/DOWN Counter 400
Successive Approximation Convertor 338
Supercomputers 363
Super Twisted Nematic LC 308
Switch Tail Ring Counters 246
Symbol of T Flip-flop 209
Synchronous Sequential Circuits 190
Synchronous UP/DOWN Counter 268
System Software 377

T

T Flip-flop 209
The 74LS195A 4-bit Parallel Access Shift Register 241
The Exclusive-OR 55
The Race-around Condition 207
Third Generation 359
characteristics of third generation computers 359
Three Input AND Gate 47
Three Input OR Gate 49
Time Delay 134, 251
time delay circuit using SISO register 251
Totem Pole (TTL) 151
Tracking ADC Circuit 338
Transformation of POS to SOP 106
Transformation of Sop to Pos 104
Transistors 358
Transistor–Transistor Logic (TTL) 150
disadvantage of single output stage TTL 151
multi-emitter TTL gate 150
TTL gate 150
working of 150
Troubleshooting in Digital Devices 157
Truth table for 3-input NAND gate 52
Truth table for 3-input NOR gate 54
Truth table for a 2-input OR gate 48
Truth table for NOT gate 50
Truth table of 2-input NOR gate 54
Truth table of a 2-input NAND gate 52
Twisted Ring Counters 246
Two's Complement 20
Types of LCDs 306

U

Ultra Large Scale Integration (ULSI) 132
Unipolar Digital Signal 2
Universal Asynchronous Receiver Transmitter (UART) 251
Universal Gates 50
Universal Shift Register 244
UP/DOWN Counter 264
UP/DOWN Counter Using IC 74190 400
UP/DOWN Counter Using IC 74193 401

V

Vacuum Tubes (VT) 301, 358
Very Large Scale Integration (VLSI) 132
Video RAM (VRAM) and Window DRAM (WRAM) 290

W

Winchester Disks 367
Wire Delay 134
Word Expansion 413
Working of LCD with a Twisted Nematic Crystal 307
Workstations 364
Write Enable (/WE) 289

X

XNOR 56
XOR Gate 55
XOR Gate Equivalent Symbol 55
XOR Gate Logic Symbol 55